KFKI-75-2

NÉMETH G.

PRAKA PROGRAMCSOMAG

Hungarian Academy of Sciences

CENTRAL
RESEARCH
INSTITUTE FOR
PHYSICS

BUDAPEST

KFKI-75-2

PRAKA PROGRAMCSOMAG

Németh G.
Számitástechnikai Főosztály
Központi Fizikai Kutató Intézet, Budapest

ISBN 963 371 001 4

KIVONAT

A jelen reportban a PRAKA programcsomag leirását ismertetjük. E programcsomag numerikus /trigonometrikus, polinom vagy racionális tört tipusu/ közelitések meghatározására alkalmas programokat tartalmaz.

АННОТАЦИЯ

В настоящей статье дается описание пакета программ ПРАКА. Пакет содержит подпрограммы для численного определения приближенных выражений тригонометрического, многочленного и рационального типа.

ABSTRACT

In this report the description of the PRAKA program package is given. The package contains programs to generation of numerical approximations in trigonometric, polynomial or rational form.

1. Bevezetés

A PRAKA programcsomag polinom és racionális approximációkat készitő algoritmusok gyüjteménye.

Ismeretes, hogy számológépeken függvények kiszámitásának egyik legkézenfekvőbb, leggyorsabb módszere közelitések használata. A közelitő kifejezés formája /a vizsgálandó függvény természetétől függően/ lehet polinom, racionális törtkifejezés /lánctört/ vagy ezek bonyolult kombinációja. E problémakör irodalmát tekintve csak Ralston [1] könyvére utalunk /ott részletes irodalmi hivatkozás található/.

A PRAKA szubrutinjai az approximációs feladatok megoldására szolgáló valamely algoritmus realizálásai. A több lehetséges /ismeretes/ módszer közüli választás az egyszerüség, könnyebb érthetőség szempontja szerint történt /ezért nem állitjuk, hogy minden esetben a feladat megoldásának legjobb módszere lett beprogramozva/.

A PRAKA szubrutinjai FORTRAN IV nyelven vannak megirva dupla preciziós változók segitségével és ezért esetenként nagypontosságu 17 decimális jegyü közelitések meghatározására is alkalmasak.

A felhasználás módja a következő. A felhasználó approximációs problémájának megoldásakor aktivizálja a csomag szubrutinjait, aképpen, hogy megirja a főszervező /master/ programot. Ez a program gondoskodik a szükséges adatbevitelről és/vagy kivitelről, gondoskodik továbbá a PRAKA szükséges szubrutinjainak hivásáról, a nyert eredmények pontosságának vizsgálatáról és egyéb teendőkről. Célszerü a főprogramot FORTRAN nyelven irni. A PRAKA szubrutinjainak matematikai leirása /müködési elv ismertetése FORTRAN nyelvü programlistája/, hivási módja, stb a 2. fejezetben lesz részletesen tárgyalva.

A PRAKA szubrutinjai segitségével az alábbi feladatok megoldása végezhető el:

a/ Interpoláció. Táblázatosan adott függvény polinom, racionális vagy trigonometrikus interpolációja, továbbá Csebisev tipusu minimax interpoláció /abszolut és relativ hiba minimalizálásával/.

b/ Ekonomizáció. Taylor sorával adott függvény polinom és racionális közelitéseinek meghatározása közvetlen Csebisev sorba való átirással Lanczos, Hornecker, ill Remez módszerével.

c/ Racionális tört kifejtések; Taylor sorával adott függvény racionális tört kifejtése Viskovatoff, Wall, Pade, Maehly módszerével valamint a Q.D. algoritmus alapján.

d/ Hipergeometriai tipusu függvények Csebisev sorfejtése. E szubrutinok a sorfejtési együtthatókat számitják ki; a sorfejtés közönséges polinommá konvertálható vagy közvetlenül felhasználható közelitő kifejezésként.

e/ Polinomokkal való müveletek /és egyéb segédprogramok/. A közelitések készitése során a probléma természetének megfelelően bizonyos segédprogramok alkalmazására van szükség. Például a Pade approximációk meghatározásához lineáris egyenletrendszert megoldó program szükséges stb.

A PRAKA szubrutinjainak fenti csoportositása nem jelent különösebb megkötöttséget, korlátozást. Sőt a FORTRAN nyelv szabályait betartva e csoportok szükithetők szubrutinok elhagyásával, vagy bővithetők uj szubrutinok beépitésével. Egyes különösen komplikált függvények esetén, ha a futtatás eredménytelen volt, további segédprogramok megirása és beépitése lehet szükséges. Sőt sorozatos, rutinjellegü feladatok megoldása esetén kiirási formátum generáló segédprogram irása célszerü /a fő szervező program egyszerüsitése érdekében/. Hasonlóan ajánlható még, hogy célszerü az eredmények analizálása céljából u.n. diagnosztikai segédprogramokat késziteni, ugyanis a PRAKA szubrutinjai nem adnak hibajelzést, ha a szubrutint helytelenül hivtuk, ha a szubrutin által számitott eredmény a gépi számábrázolási terjedelmén kivül esik, ha a megoldandó egyenletrendszer determinánsa zérus stb.

A PRAKA szubrutinjai eredetileg ICL FORTRAN-ban lettek megirva és kipróbálva. Az R-20 számológépre való átirás a FORTRAN IV nyelven néhány az algoritmusok lényegét nem érintő korlátozás segitségével történt, de az uj változat az ICL gépen is futtatható/ vagy más FORTRAN IV forditóprogrammal rendelkező gépen/.

2. A PRAKA programcsomag leirása.

a/ Interpoláció

Táblázatosan adott függvény interpolációját az alábbi POLINT, RACINT, SININT, COSINT, FURINT, ORTINT, CSEINT, MAXINT, REMINT szubrutinokkal végezhetjük el.

a1/ A POLINT program.

LEIRÁS

Adott N (x_i, f_i) pontban / x_i alappont, f_i függvényérték / interpoláló

$$P(x) = \sum_{j=0}^{N-1} a_j x^j$$

polinom együtthatóit a_j $(j=0,1,\dots,N-1)$ határozza meg. A számitás során a program a $P(x)$ polinom Euler előállitását

$$P(x) = e_0 + e_1(x-x_1) + e_2(x-x_1)(x-x_2)$$

határozza meg, majd azt átirja közönséges polinomba.

A FORTRAN lista:

```
      SUBROUTINE POLINT(A,F,N,E)
      DOUBLE PRECISION A,F,E
      DIMENSION A(N),F(N),E(N)
      DO 1 I=1,N
1     E(I)=F(I)
      DO 2 J=2,N
      R=E(J-1)
      T=A(J-1)
      DO 2 K=J,N
2     E(K)=(E(K)-R)/(A(K)-T)
      N1=N-1
      DO 3 J=1,N1
      NJ=N-J
      R=A(NJ)
      DO 3 K=NJ,N1
3     E(K)=E(K)-R*E(K+1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK,

CALL POLINT(A, F, N, E)

A,F,E N méretü tömbök ,

A tartalma: alappontok $x_1, \dots, x_N$ /bemenő/,

B tartalma: függvényértékek $f_1, \dots, f_N$ /bemenő/,

E tartalma: a P polinom a_j / $j=0,1,\dots,N-1$/ együtthatói /kimenő/,

N tartalma: az együtthatók száma/ bemenő /.

a2/ A RACINT program

LEIRÁS

Adott N (x_i, f_i) pontban / x_i alappont, f_i függvényérték $i=1,2,\dots,N$ / interpoláló

$$P(x) = e_1 + e_2(x-x_1) + \dots + e_{\ell-m+1}(x-x_1)\dots(x-x_{\ell-m}) + \\ +(x-x_1)\dots(x-x_{\ell-m+1}) \cfrac{1}{e_{\ell-m+2} + \cfrac{x-x_{\ell-m+2}}{e_{\ell-m+2} + \dots + \cfrac{x-x_{\ell+m}}{e_{\ell+m+1}}}} .$$

racionális tört kifejezés e_i $i=1,2,\dots,$ $N=\ell+m+1$ együtthatóit határozza meg.

A program a számitást az alábbi rekurzióval végzi.

$$a_{k0}=f_{k+1}\,,\quad k=0,1,\dots,\ell-m,$$

$$a_{ki}=\frac{a_{k,i-1}-a_{i-1,i-1}}{x_k-x_{i-1}}\,,\quad i=1,2,\dots,k,$$

$$a_{ii}=e_{i+1}\,,\quad i=0,1,\dots\ \ell-m,$$

$$f_k^*=\frac{(x_{k+1}-x_1)(x_{k+1}-x_2)\dots(x_{k+1}-x_{\ell-m+1})}{f_{k+1}-E(x_{k+1})}\,,\quad k=\ell-m+2,\dots,\ell+m+1,$$

$$b_{ki}=\frac{x_{k+1}-x_i}{b_{k,i-1}-b_{i-1,i-1}}\,,\quad b_{k0}=f_{k+1}^*\,,\quad i=1,2,\dots,2m,$$

$$b_{ii}=e_{i+\ell-m+1}\,,\quad i=0,1,\dots,2m.$$

$E(x)$-ℓ a $P(x)$ racionális kifejezés polinom részét jelöltük.

A FORTRAN lista:

```
      SUBROUTINE RACINT(R,A,E,N,M)
      DOUBLE PRECISION R,E,A,T,S,W
      DIMENSION R(N),A(N),E(N)
      NB=N-2*M
      DO 1 I=1,N
1     E(I)=R(I)
      IF(NB-1) 6,6,9
9     DO 2 J=2,NB
      S=E(J-1)
      T=A(J-1)
      DO 2 K=J,NB
2     E(K)=(E(K)-S)/(A(K)-T)
      N2=2*M
6     DO 3 I=1,N2
      NI=NB+I
      W=A(NI)
      T=W-A(NB)
      S=E(NB)
      IF(NB-1) 7,7,8
8     NB1=NB-1
      DO 4 J=1,NB1
      J1=NB-J
      T=T*(W-A(J1))
4     S=S*(W-A(J1))+E(J1)
7     CONTINUE
3     E(NI)=T/(E(NI)-S)
      DO 5 J=2,N2
      NB1=NB+J-1
      S=E(NB1)
      T=A(NB1)
      DO 5 K=J,N2
      NB2=NB+K
5     E(NB2)=(A(NB2)-T)/(E(NB2)-S)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL RACINT (R, A, E, N, M)

R,A,E N méretü tömbök $N = L + M + 1$, $L \geq M$,

R tartalma: f_i függvényértékek /bemenő/ ,

A tartalma: x_i alappontok /bemenő/,

E tartalma: e_i együtthatók /kimenő/,

N tartalma: az együtthatók száma /bemenő/,

M tartalma: a nevező együtthatók száma/bemenő/.

Meg kell jegyezni, hogy a feladatnak nincs mindig /bármely (x_i, f_i) pontrendszer esetén/ megoldása. Az elvi kritériumot, mely megállapitja a megoldás létezését, a program nem vizsgálja. Ha a futás sikertelen volt valamely L és M aktuális értékkel, lehetséges más L és M választással a feladatot megoldani.

Az e_i együtthatókkal megadott $P(x)$ racionális kifejezést a RENDZ2 szubrutin tudja átrendezni a szokásosabb alaku

$$P^*(x) = \frac{\sum_{i=0}^{L} a_i x^i}{\sum_{i=0}^{M} b_i x^i}, \quad b_0 = 1,$$

racionális törtté.

a3/ A SININT program.

LEIRÁS

Az adott N $x_k = \frac{k\pi}{N+1}$, $k=1,2,\dots,N$ pontbeli függvényértékekkel a

$$T(x) = \sum_{j=1}^{N} c_j \sin jx$$

trigonometrikus közelités c_j együtthatóit határozza meg a program.

A számitást a

$$c_j = \frac{2}{N+1} \sum_{l=1}^{N} f_l \sin l x_j,$$

képleten az alábbi rekurziós eljárással hajtja végre:

$$b_{N+1} = b_{N+2} = 0,$$

$$b_l = f_l + 2\cos x_j \cdot b_{l+1} - b_{l+2}, \quad l = N, N-1, \dots, 1,$$

$$C_j = \frac{2}{N+1} \sin x_j \cdot b_1 .$$

A FORTRAN lista:

```
      SUBROUTINE SININT(F,E,N)
      DOUBLE PRECISION F,E,S,C,W,P,T,B
      DIMENSION F(N),E(N),W(5)
      P=3.14159265358979D-Ø
      W(1)=2.ØD-Ø/(N+1)
      W(2)=DCOS(P/(N+1))
      W(3)=DSIN(P/(N+1))
      W(4)=1.ØD-Ø
      W(5)=Ø.ØD-Ø
      DO 1 I=1,N
      S=W(2)*W(4)-W(3)*W(5)
      W(5)=W(2)*W(5)+W(3)*W(4)
      W(4)=S
      B=2*S
      S=Ø.ØD-Ø
      C=F(N)
      N1=N-1
      DO 2 L=1,N1
      NL=N-L
      T=F(NL)+B*C-S
      S=C
    2 C=T
    1 E(I)=W(1)*C*W(5)
      RETURN
      END
```

A PROGRAM HIVÁSA, PARAMÉTEREK

CALL SININT (F, E, N)

F,E N méretü tömbök,

F tartalma: f_k függvényértékek az $x_k = \frac{k\pi}{N+1}$ pontokon /bemenő/

E tartalma: a C_k együtthatók /kimenő/.

N tartalma: az együtthatók száma/bemenő/,

HIVOTT SZUBRUTINOK: DSIN, DCOS.

a4/ A COSINT program.

LEIRÁS

Az adott N alapponthoz $x_k = \frac{(2k-1)\pi}{2N}$, $k=1,2,\dots,N$ a függvényértékek segitségével $f_k = f(x_k)$ meghatározza a

$$T(x) = \frac{1}{2}c_0 + \sum_{j=1}^{N-1} c_j \cos jx$$

alaku interpolációs polinom együtthatóit c_j-t, $j=0,1,\dots,N-1$.

Az együtthatókat a

$$c_k = \frac{2}{N}\sum_{j=1}^{N} f_j \cos kx_j$$

képlet alapján az alábbi rekurziós képletek segitségével számitja ki:

$$b_{N+1} = b_{N+2} = 0,$$

$$b_\ell = f_\ell + 2\cos j\frac{\pi}{N}\cdot b_{\ell+1} - b_{\ell+2},$$

$$\ell = N, N-1, \dots, 1, 0,$$

$$c_j = \frac{2}{N}(b_1 - b_2)\cos j\frac{\pi}{2N}.$$

A FORTRAN lista:

```
      SUBROUTINE COSINT(F,E,N)
      DOUBLE PRECISION F,E,W,S,C,T,P,R
      DIMENSION F(N),E(N),W(5)
      P=3,141592653589793D-Ø
      W(1)=2,ØD-Ø/N
      W(4)=DCOS(P/2/N)
      W(5)=-DSIN(P/2/N)
      W(2)=W(4)
      W(3)=-W(5)
      DO 1 I=1,N
      S=W(2)*W(4)-W(3)*W(5)
      W(5)=W(2)*W(5)+W(3)*W(4)
      W(4)=S
      R=4*S*S-2
      S=Ø,ØD-Ø
      C=F(N)
      N1=N-1
      DO 2 L=1,N1
      NL=N-L
      T=F(NL)+R*C-S
      S=C
    2 C=T
    1 E(I)=W(1)*W(4)*(C-S)
      E(1)=E(1)/2
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK

CALL COSINT (F, E, N)

F,E N méretü tömbök,

F tartalma: f_k függvényértékek az x_k pontokban /bemenő/,

E tartalma: a c_k együtthatók $k=0,1,\dots,N-1$ /kimenő/,

N tartalma: az együtthatók száma/bemenő/,

HIVOTT SZUBRUTINOK:DSIN,DCOS.

a5/ A FURINT program.

LEIRÁS

Adott $N=2n+1$ (x_k,f_k) $x_k=k\frac{\pi}{n}$, $k=-n,-n+1,\dots,-1,0,1,\dots,n$ pontbeli függvényérték segitségével meghatározza a

$$T(x)=\sum_{j=1}^{n-1} b_j \sin jx + \sum_{j=0}^{n} c_j \cos jx$$

trigonometrikus közelités b_j, c_j együtthatóit. A b_j és c_j együtthatók értéke a következő:

$$b_j=\frac{2}{n}\sum_{l=1}^{n-1} u_l \sin l x_j, \quad u_l=\frac{1}{2}(f_l-f_{-l}), \quad l=1,2,\dots,n-1,$$

$$c_j=\frac{2}{n}\sum_{l=0}^{n} v_l \cos l x_j, \quad v_l=\frac{1}{2}(f_l+f_{-l}), \quad l=1,2,\dots,n-1,$$

$$v_0=\frac{1}{4}f_0, \quad v_n=\frac{1}{4}f_n.$$

A b_j és c_j számokat a program rekurziv módszerrel számit-ja ki /e számitás teljesen analog a SININT valamint a CO-SINT programokban alakalmazott módszerrel/.

A FORTRAN lista:

```
      SUBROUTINE FURINT(F,E,N,R)
      DOUBLE PRECISION F,E,W,S,C,T,P,R
      DIMENSION F(N),E(N),W(5),T(6),R(N)
      P=3,14159265358979 3D-0
      M=(N-1)/2
      W(1)=2,0D-0/M
      W(2)=DCOS(P/M)
      W(3)=DSIN(P/M)
      W(4)=1,0D-0
      W(5)=0,0D-0
      R(M+1)=F(M+1)/2
      R(N)=F(N)/2
      DO 1 I=2,M
      IN=N-I+1
      R(I)=(F(IN)-F(I))/2
1     R(IN)=(F(IN)+F(I))/2
      S=R(M+1)
      C=S
      DO 2 I=1,M
      IM=I+M+1
      S=S+R(IM)
      W(4)=-W(4)
2     C=C+W(4)*R(IM)
      E(M+1)=S/M
      E(N)=C/M
      W(4)=1,0D-0
      M1=M-1
      DO 3 I=1,M1
      S=W(2)*W(4)-W(3)*W(5)
      W(5)=W(2)*W(5)+W(3)*W(4)
      W(4)=S
      T(2)=0,0D-0
      T(1)=0,0D-0
      T(4)=0,0D-0
      T(3)=R(N)
      T(5)=S*2
      M1=M-1
      DO 4 L=1,M1
      T(6)=R(L+1)+T(5)*T(1)-T(2)
      T(2)=T(1)
      T(1)=T(6)
      NL=N-L
      T(6)=R(NL)+T(5)*T(3)-T(4)
      T(4)=T(3)
4     T(3)=T(6)
      E(I)=W(1)*T(1)*W(5)
      NM=N-M
      T(4)=R(NM)+T(5)*T(3)-T(4)
      MI=M+I+1
3     E(MI)=W(1)*(T(4)-W(4)*T(3))
      E(M)=0,0D-0
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL FURINT (F, E, N, R)

F,E,R N méretü tömbök,

F tartalma: f_k függvényértékek az $x_k = k\frac{\pi}{n}$ /bemenő/ pontokban, $k=-n,-n+1,\dots,-1,0,1,\dots,n$.

E tartalma: $b_1, b_2, \dots, b_{n-1}, 0, c_0, c_1, \dots, c_n$ /kimenő/,

R tartalma: közömbös, munkatömb /be - és kimenő/.

N tartalma: az együtthatók száma/bemenő/,

HIVOTT SZUBRUTINOK:DSIN,DCOS.

a6/ Az ORTINT program.

LEIRÁS

Adva van N alappont: x_k a hozzátartozó f_k függvényértékekkel. Meghatározandó a sulyzott legkisebb négyzetek elve alapján a függvény értékeit közelitő

$$F(x) = \sum_{i=0}^{M} a_i x^i$$

polinom, az a_i együtthatók szórásaival együtt.

A számitás az alábbi képleteket használja:

$$P(x) = \sum_{j=0}^{M} c_j p_j(x), \qquad c_j = \frac{\sum_{i=1}^{N} \omega_i p_j(x_i) y_i}{\sum_{i=1}^{N} \omega_i p_j^2(x_i)}$$

$$\sum_{i=1}^{N} \omega_i p_j(x_i) p_k(x_i) = 0, \quad k \neq j; \qquad p_0(x) = 1, \quad p_1(x) = x - \alpha_1,$$

$$p_{j+1}(x) = (x - \alpha_{j+1}) p_j(x) - \beta_j p_{j-1}(x), \quad j = 1, 2, \dots, M,$$

$$\alpha_{j+1} = \frac{\sum_{i=1}^{N} \omega_i x_i p_j^2(x_i)}{\sum_{i=1}^{N} \omega_i p_j^2(x_i)}, \qquad \beta_j = \frac{\sum_{i=1}^{N} \omega_i p_j^2(x_i)}{\sum_{i=1}^{N} \omega_i p_{j-1}^2(x_i)},$$

$$R_j^2 = \left\{ \sum_{i=0}^{N} \left(f(x_i) - \sum_{\ell=0}^{j} c_\ell p_\ell(x_i) \right)^2 \omega_i \right\} / (N-j-1); \qquad \sigma_{a_k}^2 = R_j^2 \sum_{\ell=k}^{j} \frac{\beta_{\ell k}^2}{\gamma_\ell},$$

$$p_j(x)=\sum_{r=0}^{j}\beta_{jr}x^r\ ,\qquad \gamma_j=\sum_{i=1}^{N}\omega_i\,p_j^2(x_i)\ .$$

A FORTRAN lista:

```
      SUBROUTINE ORTINT(F,X,S,W,N,E,L,B,C,D)
      DOUBLE PRECISION F,X,S,W,T,Z,Q,Y,E,B,C,D
      DIMENSION F(N),X(N),S(N),W(3),E(L),B(N),C(N),D(N)
      T=0.0D-0
      Z=T
      Y=T
      Q=T
      DO 1 I=1,N
      B(I)=0.0D-0
      C(I)=0.0D-0
      D(I)=0.0D-0
      T=T+F(I)*S(I)
      Z=Z+S(I)
      Y=Y+X(I)*S(I)
1     Q=Q+F(I)**2*S(I)
      B(1)=1.0D-0
      D(1)=B(1)/Z
      E(1)=T/Z
      E(3)=Q-T*T/Z
      E(2)=E(3)/Z/(N-1)
      C(1)=-Y/Z
      C(2)=B(1)
      W(1)=-C(1)
      Y=0.0D-0
      T=Y
      Q=Y
      DO 2 I=1,N
      W(2)=X(I)-W(1)
      Y=Y+W(2)**2*S(I)
      T=T+W(2)**2*S(I)*X(I)
2     Q=Q+W(2)*F(I)*S(I)
      E(5)=Q/Y
      E(4)=E(1)-W(1)*E(5)
      D(1)=D(1)+C(1)**2/Y
      D(2)=D(2)+C(2)**2/Y
      E(8)=E(3)-Q*Q/Y
      E(6)=E(8)/(N-2)*D(1)
      E(7)=E(8)/(N-2)*D(2)
      DO 3 J=2,M
      W(1)=Y/Z
      Z=Y
      W(2)=T/Z
      C(J+1)=1.0D-0
      J1=J-1
      DO 4 K=1,J1
      JK=J-K
      W(3)=C(JK)-W(2)*C(JK+1)-W(1)*B(JK+1)
      B(JK+1)=C(JK+1)
4     C(JK+1)=W(3)
      W(3)=C(1)
      C(1)=-W(2)*C(1)-W(1)*B(1)
      B(1)=W(3)
      Y=0.0D-0
      T=Y
      Q=Y
      DO 5 L1=1,N
      W(3)=C(J+1)
```

```
      DO 6 K=1,J
      JK=J-K+1
6     W(3)=W(3)*X(L1)+C(JK)
      Y=Y+W(3)**2*S(L1)
      T=T+W(3)**2*X(L1)*S(L1)
5     Q=Q+S(L1)*W(3)*F(L1)
      W(3)=Q/Y
      N7=J*(J+2)
      N8=J*(J+3)+1
      N9=(J+1)*(J+3)
      N10=2*J+1
      E(N9)=E(N7)-Q*Q/Y
      DO 7 K=1,J
      N11=N7+K
      N12=N7-N10+K
      E(N11)=E(N12)+W(3)*C(K)
      D(K)=D(K)+C(K)**2/Y
      N13=N8+K
7     E(N13)=D(K)*E(N9)/(N-J-1)
      E(N11+1)=W(3)*C(J+1)
      D(J+1)=C(J+1)**2/Y
      E(N13+1)=D(J+1)*E(N9)/(N-J-1)
3     CONTINUE
      M1=M+1
      DO 8 I=1,M1
      I1=I*(I+2)
8     E(I1)=E(I1)/(N-1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL ORTINT(F,X,S,N,M,E,L,B,C,D)

F,X,S,B,C,D N méretü, E $L=(M+1)(M+2)$ méretü tömbök,

F tartalma: $f_1, \ldots, f_N$ függvényértékek az $x_1, \ldots, x_N$ pontokban. /bemenő/,

X tartalma: $x_1, \ldots, x_N$ pontok /bemenő/,

S tartalma: $\omega_1, \ldots, \omega_N$ sulyok az $x_1, \ldots, x_N$ pontokban /bemenő/,

N az adatok száma /bemenő/,

M a közelitő polinom maximális foka /bemenő/,

E tartalma: az eredmény; a közelitő polinomok együtthatói szórásaikkal együtt az alábbi sorrendben:

$$a_0, \sigma_{a_0}^2, R_0^2;\ a_0, a_1, \sigma_{a_0}^2, \sigma_{a_1}^2, R_1^2;$$

$$a_0, a_1, a_2, \sigma_{a_0}^2, \sigma_{a_1}^2, \sigma_{a_2}^2, R_2^2; \ldots$$

$$a_0, a_1, \ldots, \sigma_{a_0}^2, \ldots \sigma_{a_M}^2, R_M^2,$$ /kimenő/

$\left.\begin{array}{l} B \\ C \\ D \end{array}\right\}$ munkatömbök, tartalmuk közömbös /be-, kimenő/,

a7/ A CSEINT program.

LEIRÁS

Adott $N=n+2$, $x_k=\frac{k\pi}{n+2}$, $k=0,1,\dots,n+1$ alappontu függvényértékekből $f_0, f_1, \dots, f_{n+1}$ meghatározza a

$$T(x)=\sum_{j=0}^{n}(-1)^j c_j \cos jx$$

trigonometrikus közelitést.

A c_j együtthatók formulája

$$c_j=\frac{2}{n+1}\sum_{l=0}^{n+1}{}'' f_l \cos l x_j.$$

A közelités hibája

$$h=\frac{1}{n+1}\sum_{l=0}^{n+1}{}''(-1)^l f_l.$$

A képletekben szereplő két vessző azt jelenti, hogy az első és utolsó tag az összegben $1/2$ szorzóval számitandó.

A FORTRAN lista:

```
      SUBROUTINE CSEINT(F,N,E)
      DOUBLE PRECISION F,E,W,P,S,T,C,Z,A,B,R,V,U
      DIMENSION F(N),E(N)
      P=3,14159265358979ЗD-Ø
      F(1)=F(1)/2
      F(N)=F(N)/2
      W=(2,ØD-Ø)/(N-1,Ø)
      S=DCOS(P/(N-1))
      T=DSIN(P/(N-1))
      C=1,ØD-Ø
      Z=Ø,ØD-Ø
      A=C
      DO 1 I=1,N
      R=Ø,ØD-Ø
      V=F(N)
      B=2*C
      N1=N-1
      DO 2 L=1,N1
      U=F(N-L)+B*V-R
      R=V
2     V=U
```

```
      E(I)=W*A*(V-C*R)
      V=S*C-T*Z
      Z=S*Z+T*C
      C=V
1     A=-A
      E(1)=E(1)/2
      E(N)=E(N)/2
      F(1)=F(1)*2
      F(N)=F(N)*2
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL CSEINT (F,N,E)

F,E N méretü tömbök,

F tartalma: $f_1, f_2, \dots, f_{n+1}$, $N=n+1$ függvényértékek /bemenő/,

E tartalma: az eredmény együtthatói az alábbi sorrendben:

$c_0, c_1, \dots, c_n$, h /kimenő/,

N tartalma: az együtthatók száma/bemenő/,

HIVOTT SZUBRUTINOK:DSIN,DCOS.

a8/ A MAXINT program.

LEIRÁS

Adott $N=n+2$ $x_1, x_2, \dots x_{n+2}$ alappontból és a hozzátartozó $f_1, f_2, \dots, f_{n+2}$ függvényértékből meghatározza azt a

$$P(x) = \sum_{l=0}^{n} a_l x^l$$

polinomot, amely az x_i helyen $f_i + (-1)^i h$ értéket vesz fel /minimax Csebisev közelités pontsorozaton/. A $P(x)$ polinomot az Euler féle alakban határozza meg:

$$P(x) = f_1 - h + \sum_{l=2}^{n+1} \left(\Delta^l f_k - h \Delta^l (-1)^k\right)(x-x_1)\dots(x-x_{l-1})$$

majd visszarendezi algebrai polinommá. A h értéke

$$h = \left(\Delta^{n+2} f_k\right) / \left(\Delta^{n+2} (-1)^k\right).$$

A FORTRAN lista:

```
      SUBROUTINE MAXINT(F,X,N,E1,E2)
      DOUBLE PRECISION F,X,E1,E2,W,S,T
      DIMENSION F(N),X(N),E1(N),E2(N)
      E1(1)=F(1)
      E2(1)=1.0D-0
      T=1.0D-0
      DO 1 I=2,N
      T=-T
      E1(I)=F(I)
    1 E2(I)=T
      DO 2 J=2,N
      S=X(J-1)
      W=E1(J-1)
      T=E2(J-1)
      DO 2 K=J,N
      E1(K)=(E1(K)-W)/(X(K)-S)
    2 E2(K)=(E2(K)-T)/(X(K)-S)
      E1(N)=E1(N)/E2(N)
      N1=N-1
      DO 3 I=1,N1
    3 E1(I)=E1(I)-E1(N)*E2(I)
      N1=N1-1
      DO 4 J=1,N1
      J1=N-J-1
      W=X(J1)
      DO 4 K=J1,N1
    4 E1(K)=E1(K)-W*E1(K+1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL MAXINT(F,X,N,E1,E2)

F,X,E1,E2 N méretü tömbök,

F tartalma: $f_1, f_2, \dots, f_{n+2}$ függvényértékek /bemenő/,

X tartalma: $x_1, x_2, \dots, x_{n+2}$ alappontok /bemenő/,

E1 tartalma: az eredmény együtthatók az alábbi sorrendben:

$a_0, a_1, \dots, a_n, h$ /kimenő/,

E2 munkatömb tartalma közömbös /be-, kimenő/.

N tartalma: az együtthatók száma/bemenő/.

a9/ A REMINT program.

LEIRÁS

Adott $N = n+2$ $x_1, x_2, \dots, x_{n+2}$ alappontból és a hozzátartozó $f_1, f_2, \dots, f_{n+2}$ függvényértékből meghatározza azt a

$$P(x) = \sum_{\ell=0}^{n} a_\ell x^\ell$$

polinomot, amely az x_i helyen az $f_i(1+(-1)^i h)$ értéket veszi fel /relativ minimax közelités pontsorozaton/.

A $P(x)$ polinomot a program az Euler féle alakban

$$P(x) = f_1(1-h) + \sum_{\ell=2}^{n+1} \left(\Delta^\ell f_k - h\Delta^\ell[f_k(-1)^k]\right)(x-x_1)\dots(x-x_{\ell-1})$$

határozza meg, majd visszarendezi algebrai polinommá.

A h értéke

$$h = \left(\Delta^{n+2} f_k\right) / \left(\Delta^{n+2}[f_k(-1)^k]\right).$$

A FORTRAN lista:

```
      SUBROUTINE REMINT(F,X,N,E1,E2)
      DOUBLE PRECISION F,X,E1,E2,W,S,T
      DIMENSION F(N),X(N),E1(N),E2(N)
      E1(1)=F(1)
      E2(1)=F(1)
      T=1,0D-0
      DO 1 I=2,N
      T=-T
      E1(I)=F(I)
1     E2(I)=T*F(I)
      DO 2 J=2,N
      S=X(J-1)
      W=E1(J-1)
      T=E2(J-1)
      DO 2 K=J,N
      E1(K)=(E1(K)-W)/(X(K)-S)
2     E2(K)=(E2(K)-T)/(X(K)-S)
      E1(N)=E1(N)/E2(N)
      N1=N-1
      DO 3 I=1,N1
3     E1(I)=E1(I)-E1(N)*E2(I)
```

```
      N1=N1-1
      DO 4 J=1,N1
      J1=N-J-1
      W=X(J1)
      DO 4 K=J1,N1
4     E1(K)=E1(K)-W*E1(K+1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL REMINT(F,X,N,E1,E2)

F,X,E1,E2 N méretü tömbök,

F tartalma: $f_1, f_2, \dots, f_{n+2}$ függvényértékek /bemenő/,

X tartalma: $x_1, x_2, \dots, x_{n+2}$ alappontok /bemenő/,

E1 tartalma: az eredmény együtthatói az alábbi sorrendben:

$a_0, a_1, \dots, a_n, h$ /kimenő/,

E2 tartalma közömbös /munkatömb/, /be-, kimenő/,

N tartalma: az együtthatók száma/bemenő/.

b/ Ekonomizáció.

Taylor sorával adott függvény polinom és racionális közelitésének meghatározását elvégezhetjük, közvetlen Csebisev sorba való átirással, Lanczos módszerével, Hornecker módszerével ill. Remez módszerével a TAYCSE,CSETAY,SZTACS,SZCSTA,TRUNC1,TRUNC2, PSEMAE,HORNEC,MINMAX,MINRAC szubrutinok segitségével.

b1/ A TAYCSE program.

LEIRÁS

Taylor részletösszegből Csebisev részletösszeget készit a $(0,1)$ intervallumra vonatkozóan az alábbi azonosság alapján:

$$\sum_{l=0}^{n} a_l x^l = \sum_{l=0}^{n} b_l T_l^*(x) .$$

A b_k számokat az alábbi képlet alapján számitja ki:

$$b_0=\sum_{k=0}^{n} a_k \varrho_k\,, \qquad \varrho_0=1\,, \qquad \varrho_k=\frac{k-\frac{1}{2}}{k}\,\varrho_{k-1}\,, \qquad k=1,2,\dots,$$

$$b_\ell=\frac{2}{4^\ell}\sum_{k=0}^{N-\ell} a_{k+\ell}\,\varrho_k^{(\ell)} \quad ; \quad \varrho_0^{(\ell)}=1,$$

$$\varrho_k^{(\ell)}=\frac{(2k+2\ell-1)(k+\ell)}{2k(k+2\ell)}\,\varrho_{k-1}^{(\ell)}\,, \qquad k=1,2,\dots$$

A FORTRAN lista:

```
      SUBROUTINE TAYCSE(T,C,N)
      DOUBLE PRECISION T,C,A,B,G,D
      DIMENSION T(N),C(N)
      A=2.0D-0
      B=A
      DO 1 I=1,N
      G=1.0D-0
      D=T(I)
      IF(I-N) 4,2,4
    4 L=N-I
      DO 3 J=1,L
      G=G*(B*(J+I)-3)*(J+I-1)/B/(J+B*(I-1))/J
    3 D=D+T(I+J)*G
      IF(I-1) 5,5,2
    5 C(1)=D
      GO TO 1
    2 C(I)=A*D
    1 A=A/4
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL TAYCSE(T, C, N)

T,C N méretü tömbök,

T tartalma: $a_0, a_1, \ldots, a_n$ Taylor együtthatók /bemenő/,

C tartalma: $b_0, b_1, \ldots, b_n$ Csebisev együtthatók /kimenő/,

N=n+1 a polinom együtthatóinak száma /bemenő/.

b2/ A CSETAY program.

LEIRÁS

Csebisev részletösszegből Taylor részletösszeget készit a $(0,1)$ intervallumra vonatkozóan az alábbi azonosság alapján:

$$\sum_{k=0}^{n} a_k T_k^*(x) = \sum_{k=0}^{n} b_k x^k .$$

A b_k számokat a következő képlet alapján számitja ki:

$$b_k = \sum_{l=k}^{n} a_l c_k^{(l)}$$

ahol c_k^l a Csebisev polinom együtthatója.

$$T_l^*(x) = \sum_{j=0}^{l} c_j^{(l)} x^j .$$

A FORTRAN lista:

```
      SUBROUTINE CSETAY(C,T,N)
      DOUBLE PRECISION C,T,A,S
      DIMENSION C(N),T(N)
      COMMON/COEFF/A(496)
      DO 1 I=1,N
      S=0.D0-0
      DO 2 J=I,N
      J1=((J-1)*J)/2+I
2     S=S+A(J1)*C(J)
1     T(I)=S
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL CSETAY (C, T, N)

C,T N méretü tömbök,

C tartalma: $a_0, a_1, \dots, a_n$ Csebisev együtthatók /bemenő/,

T tartalma: $b_0, b_1, \dots, b_n$ Taylor együtthatók /kimenő/,

N=n+1 a polinom együtthatóinak száma, $n \leq 30$ /bemenő/.

A szubrutin használja a COEFF nevü COMMON mező A /496 méretü/ tömbjét, amelynek az első N $T_k^*(x)$ polinom együtthatóját kell tartalmaznia. Ezért fontos, hogy a CSETAY szubrutin első hivása előtt /időrendileg/ végrehajtódjon a CALL EGYUTT(N) utasitás, amely hatására az EGYUTT nevü szubrutin /lásd e13/-t/ kitölti az A COMMON tömböt.

b3/ A SZTACS program.

LEIRÁS

Taylor részletösszegből Csebisev részletösszeget készit a $(-1,1)$ intervallumra vonatkozóan az alábbi azonosság alapján:

$$\sum_{l=0}^{n} a_l x^l = \sum_{l=0}^{n} b_l T_l(x)$$

A b_l számokat a következő képletek segitségével számitja ki:

$$b_0 = \sum_{k=0}^{2k \leq n} a_{2k} \varrho_k ,$$

$$\varrho_0 = 1 , \quad \varrho_k = \frac{k - 1/2}{k} \varrho_{k-1} , \quad k = 1, 2, \dots ,$$

$$b_l = \frac{2}{2^l} \sum_{k=0}^{l+2k \leq n} a_{l+2k} \varrho_k^{(l)} ,$$

$$\varrho_0^{(l)} = 1 , \quad \varrho_k^{(l)} = \frac{(l+2k)(l+2k-1)}{4k(l+k)} \varrho_{k-1}^{(l)} , \quad k = 1, 2, \dots ,$$

A FORTRAN lista:

```
      SUBROUTINE SZTACS(T,C,N)
      DOUBLE PRECISION T,C,S
      DIMENSION T(N),C(N),S(4)
      S(1)=2.0D-0
      S(2)=S(1)
      DO 1 I=1,N
      S(3)=1.0D-0
      S(4)=0.0D-0
      DO 3 J=1,N,2
      S(4)=S(4)+S(3)*T(J)
3     S(3)=S(3)*(J+1)*J/(J+1)/(J+2-I)
      IF(I-1) 10,10,2
10    C(I)=S(4)
      GO TO 1
2     C(I)=S(4)*S(1)
1     S(1)=S(1)/S(2)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SZTACS (T,C,N)

T,C N méretü tömbök,

T tartalma: $a_0, a_1, \ldots, a_n$ Taylor együtthatók /bemenő/,

C tartalma: $b_0, b_1, \ldots, b_n$ Csebisev együtthatók /kimenő/,

N=n+1 a polinom együtthatóinak száma /bemenő/.

b4/ A SZCSTA program.

LEIRÁS

Csebisev részletösszegből Taylor részletösszeget készit a $(-1,1)$ intervallumra az alábbi azonosság alapján

$$\sum_{k=0}^{n} a_k T_k(x) = \sum_{k=0}^{n} b_k x^k.$$

A b_k számokat a következő képlet alapján számolja ki.

$$b_k = \sum_{j=0}^{k+2j\le N} a_{k+2j}\, c_j^{(k+2j)}$$

ahol $c_k^{(l)}$ a Csebisev polinom együtthatója:

$$T_k(x) = \sum_{l=0}^{[\frac{k}{2}]} c_l^{(k)} x^{n-2l}.$$

A FORTRAN lista:

```
      SUBROUTINE SZCSTA(C,T,N)
      DOUBLE PRECISION C,T,A,S
      DIMENSION C(N),T(N)
      COMMON/COEFF/A(496)
      DO 1 I=1,N
      S=0.0D-0
      DO 2 J=I,N,2
      JB=((J-1)*J)/2+I
    2 S=S+A(JB)*C(J)
    1 T(I)=S
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SZCSTA (C,T,N)

C;T N méretü tömbök,

C tartalma: $a_0, a_1, \dots, a_n$ Csebisev együtthatók /bemenő/,

T tartalma: $b_0, b_1, \dots, b_n$ Taylor együtthatók /kimenő/,

N=n+1 a polinom együtthatóinak száma; $n \le 30$ /bemenő/.

A szubrutin használja a COEFF nevü COMMON mező A /496 méretü/ tömbjét, amelynek az első N $T_k(x)$ polinom együtthatóit kell tartalmaznia. Ezért fontos, hogy az SZCSTA szubrutin első hivása előtt /időrendileg/ végrehajtódjon a CALL SZEGYU (N) utasitás, amely hatására a SZEGYU nevü szubrutin /lásd el4/-t/ kitölti az A COMMON tömböt.

b5/ A TRUNC1 program.

LEIRÁS

Polinom alacsonyabbfoku polinommá való csonkitása Lánczos módszerével. Adott $(0,1)$-ben a

$$P(x)=\sum_{i=0}^{n} a_i x^i$$

polinom, amely az $f(x)$ függvényt E hibával közeliti. Meghatározzuk azt az "m" foku $Q(x)$ polinomot

$$Q(x)=\sum_{i=0}^{m} b_i x^i$$

amely az $f(x)$ függvényt H ($\geq E$) hibával közeliti. A megoldás módszere az, hogy az x^n hatványt legjobb közelitésével helyettesitjük:

$$x^n \sim -2^{-2n+1} \sum_{j=0}^{n-1} C_j^{(n)} x^j$$

ahol

$$C_j^{(n)} = (-1)^{n+j} \frac{4^j n \, \Gamma(n+j)}{\Gamma(2j+1)\,\Gamma(n-j+1)}.$$

Ennek során a közelités hibája E megnövekszik $a_n 2^{-2n+1}$-el. A helyettesitési eljárás addig folytatódik, amig a hiba el nem érte a H értéket.

A FORTRAN lista:

```
      SUBROUTINE TRUNC1(C,E,N,M,H,R)
      DOUBLE PRECISION C,T,E,H,R
      DIMENSION C(N),R(N)
      M=N
1     R(M)=-C(M)
      DO 2 I=2,M
      J=M-I+1
2     R(J)=-R(J+1)*J*(J-0.5D-0)/(M-J)/(M+J-2)
      T=R(1)
      IF(T) 10,3,3
10    T=-T
3     T=E+T
      IF(T-H) 11,6,6
11    E=T
      M=M-1
      DO 5 I=1,M
5     C(I)=C(I)+R(I)
      GO TO 1
6     CONTINUE
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL TRUNC1 (C, E, N, M, H, R)

C,R N méretü tömbök,

C tartalma: $a_0, a_1, \dots, a_n$ az eredeti polinom együtthatói /bemenő-, kimenő/,

Az eredmény együtthatói szintén a C tömbben lesznek az alábbi sorrendben $C(1)=b_0, \dots, C(N)=b_n$ /tehát az $a_0, a_1, \dots,$ kezdeti értékek elvesznek/.

N=n+1: a kezdeti polinom együtthatóinak száma /bemenő/,

M a végső polinom együtthatóinak száma /kimenő/,

E kezdeti közelités hibája /bemenő/,

H végső közelités hibája /bemenő/,

R munkatömb, tartalma közömbös /be-, kimenő/.

b6/ A TRUNC2 program.

LEIRÁS

Polinom alacsonyabbfoku polinommá csonkitása Lánczos módszerével. Adott a $(-1, 1)$ -ben a

$$P(x)=\sum_{i=0}^{n} a_i x^i$$

polinom, amely az $f(x)$ függvényt E hibával közeliti. Meghatározzuk azt az "m"-d foku $Q(x)$ polinomot

$$Q(x)=\sum_{i=0}^{m} b_i x^i$$

amely az $f(x)$ függvényt $H(\geq E)$ hibával közeliti. A megoldás módszere az hogy az x^n hatványt legjobb közelitésével helyettesitjük

$$x^n \sim -\sum_{i=1}^{[n/2]} a_i^{(n)} x^{n-2i}.$$

ahol

$$a_i^{(n)} = (-1)^i \frac{4^{-i}\, n\, \Gamma(n-i)}{\Gamma(n-2i+1)\,\Gamma(i+1)}$$

Ennek során a közelités hibája E megnövekszik $a_n\, 2^{-n+1}$ -ℓ.
A helyettesitési eljárás addig folytatódik, amig a hiba el nem érte a H értéket.

A FORTRAN lista:

```
      SUBROUTINE TRUNC2(C,E,N,M,H,R)
      DOUBLE PRECISION C,T,E,H,R
      DIMENSION C(N),R(N)
      M=N
1     R(M)=-C(M)
      NN=M-2
      DO 3 J=1,NN,2
      K=M-J
3     R(K-1)=-R(K+1)*K*(K-1)/(NN-1+K)/(NN+3-K)
      IF(K-2) 4,4,5
4     T=R(1)
      GO TO 2
5     T=R(2)/(M-1)
2     IF(T) 10,10,6
10    T=-T
6     E=E+T
      IF(E-H) 7,7,8
7     M=M-1
      DO 9 J=K,M,2
9     C(J-1)=C(J-1)+R(J-1)
      GO TO 1
8     RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL TRUNC2 (C, E, N, M, H, R)

C,R N méretü tömbök,

C tartalma: $a_0, a_1, \ldots, a_n$ az eredeti polinom együtthatói /bemenő-, kimenő/,

Az eredmény együtthatói szintén a C tömbben lesznek az alábbi sorrendben: $C(1)=b_0$, $C(2)=b_1, \ldots$, /tehát az $a_0, a_1, \ldots,$ kezdeti értékek elvesznek/.

N=n+1 a kezdeti polinom együtthatóinak száma /bemenő/, M a végső polinom együtthatóinak száma /kimenő/,

E a kezdeti közelités hibája /bemenő/,

H a végső közelités hibája /bemenő/,

R munkatömb, tartalma közömbös /be-,kimenő/.

b7/ A HORNEC program.

LEIRÁS

Az $f(x)$ függvény Csebisev sorfejtésének együtthatóit perturbálja. E perturbáció eredményeként a részletösszeg a függvény legjobb egyenletes megközelitését állitja elő a $(0,1)$ intervallumra vonatkozólag. Tehát

$$C(x)=\sum_{k=0}^{m} c_k T_k^*(x)$$

közelitésből

$$C^*(x)=\sum_{k=0}^{m} c_k^* T_k^*(x)$$

közelitést készit. A c_k^* együtthatókat és a közelités hibáját az alábbi képletek adják: /[3]/

Együtthatóképletek:

$$\left.\begin{aligned} c_0^* &= c_0 - \frac{c_3^2}{c_2} + c_4 + 2\frac{c_3^4}{c_2^3} - 4c_4\frac{c_3^2}{c_2^2} + 2c_5\frac{c_3}{c_2}, \\ c_1^* &= c_1 + c_3 + c_5 , \end{aligned}\right\} m = 1$$

$$\left.\begin{aligned} c_0^* &= c_0 + \frac{c_4^3}{c_3^2} - 2c_4\frac{c_5}{c_3} + c_6 , \\ c_1^* &= c_1 + c_5 - \frac{c_4^2}{c_3} + \frac{c_4^4}{c_3^3} - \frac{c_4^2 c_5}{c_3^2} - \frac{c_5^2}{c_3} + c_7 , \\ c_2 &= c_2 + c_4 - \frac{c_4^3}{c_3^2} + 2\frac{c_4 c_5}{c_3} . \end{aligned}\right\} m=2$$

Továbbá, m> 2 esetén csak az utolsó négy együttható perturbálódik, a többi változatlan.

$$c^*_{m-3} = c_{m-3} - \frac{c_{m+2}^4}{c_{m+1}^3} - \frac{c_{m+3}^2}{c_{m+1}} + 3\frac{c_{m+2}^2 c_{m+3}}{c_{m+1}^2} - 2\frac{c_{m+2}c_{m+4}}{c_{m+1}} + c_{m+5},$$

$$c^*_{m-2} = c_{m-2} + \frac{c_{m+2}^3}{c_{m+1}^2} - 2\frac{c_{m+2}c_{m+3}}{c_{m+1}} + c_{m+4},$$

$$c^*_{m-1} = c_{m-1} - \frac{c_{m+2}^2}{c_{m+1}} + c_{m+3} + 2\frac{c_{m+2}^4}{c_{m+1}^3} - 4\frac{c_{m+2}^2 c_{m+3}}{c_{m+1}^2} + 2\frac{c_{m+2}c_{m+4}}{c_{m+1}},$$

$$c^*_{m} = c_m + c_{m+2} - \frac{c_{m+2}^3}{c_{m+1}^2} + 2\frac{c_{m+2}c_{m+3}}{c_{m+1}}.$$

Hibaképletek:

$$H = c_2 + \frac{c_3^2}{c_2} + 4c_4\frac{c_3^2}{c_2^2} - 2\frac{c_3^4}{c_2^3} - 2\frac{c_3c_5}{c_2} + c_6, \quad m=1,$$

$$H = c_{m+1} + \frac{c_{m+2}^2}{c_{m+1}} + \frac{c_{m+3}^2}{c_{m+1}} + \frac{c_{m+2}^2 c_{m+3}}{c_{m+1}^2} - \frac{c_{m+2}^4}{c_{m+1}^3}, \quad m \geq 1.$$

A FORTRAN lista:

```
      SUBROUTINE HORNEC(A,N,B,H)
      DOUBLE PRECISION A,B,H
      DIMENSION A(N),B(N)
      IF(N-8) 10,10,1
10    H=A(3)+A(4)**2/A(3)+4*A(5)*(A(4)/A(3))**2
      H=H-2*A(4)**4/A(3)**3-2*A(4)/A(3)*A(6)+A(7)
      IF(H)11,11,2
11    H=-H
      GO TO 2
1     H=A(N-5)+A(N-4)**2/A(N-5)+A(N-3)**2/A(N-5)
      H=H+A(N-4)**2*A(N-3)/A(N-5)**2-A(N-4)**4/A(N-5)**3
      IF(H) 12,12,2
12    H=-H
2     IF(N-9) 13,3,4
13    B(1)=A(1)-A(4)**2/A(3)+A(5)+2*A(4)**4/A(3)**3
      B(1)=B(1)-4*A(4)**2*A(5)/A(3)**2+2*A(4)*A(6)/A(3)
      B(2)=A(2)+A(4)+A(6)
      GO TO 5
```

```
3     B(1)=A(1)+A(5)**3/A(4)**2-2*A(5)*A(6)/A(4)+A(7)
      B(2)=A(2)-A(5)**2/A(4)+A(6)+A(5)**4/A(4)**3-A(5)**2*A(6)/A(4)**2
      B(2)=B(2)-A(6)**2/A(4)+A(8)
      B(3)=A(3)+A(5)-A(5)**3/A(4)**2+2*A(5)*A(6)/A(4)
      GO TO 5
4     IF(N-10) 14,14,6
14    B(1)=A(1)-A(6)**4/A(5)**3-A(7)**2/A(5)+3*A(6)**2*A(7)/A(5)**2
      B(1)=B(1)-2*A(6)*A(8)/A(5)+A(9)
      B(2)=A(2)+A(6)**3/A(5)**2-2*A(6)*A(7)/A(5)+A(8)
      B(3)=A(3)-A(6)**2/A(5)+A(7)+2*A(6)**4/A(5)**3
      B(3)=B(3)-4*A(6)**2*A(7)/A(5)**2+2*A(6)*A(8)/A(5)
      B(4)=A(4)+A(6)-A(6)**3/A(5)**2+2*A(6)/A(7)/A(5)
      GO TO 5
6     N10=N-10
      DO 7 I=1,N10
7     B(I)=A(I)
      B(N-9)=A(N-9)-A(N-4)**4/A(N-5)**3-A(N-3)**2/A(N-5)
      B(N-9)=B(N-9)+3*A(N-4)**2*A(N-3)/A(N-5)**2
      B(N-9)=B(N-9)-2*A(N-4)*A(N-2)/A(N-5)+A(N-1)
      B(N-8)=A(N-8)+A(N-4)**3/A(N-5)**2-2*A(N-4)*A(N-3)/A(N-5)+A(N-2)
      B(N-7)=A(N-7)-A(N-4)**2/A(N-5)+A(N-3)+2*A(N-4)**4/A(N-5)**3
      B(N-7)=B(N-7)-4*A(N-4)**2*A(N-3)/A(N-5)**2+2*A(N-4)*A(N-2)/A(N-5)
      B(N-6)=A(N-6)+A(N-4)-A(N-4)**3/A(N-5)**2+2*A(N-4)*A(N-3)/A(N-5)
5     RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL HORNEC (A,N,B,H)

A,B N méretü tömbök,

A tartalma: $c_0, c_1, \ldots, c_N$ együtthatók /bemenő/,

B tartalma: a perturbált $c_0^*, c_1^*, \ldots, c_N^*$ együtthatók /kimenő/,

H tartalma: a közelités hibája/kimenő/,

N=n+7, ahol n a polinom fokszáma /igy elsőfoku polinom esetén N=8, másodfoku esetén N=9 stb, vagyis 6-l több együttható szükséges/, /bemenő/.

b8/ A PSEMAE program.

LEIRÁS

Polinom alaku legjobb közelités meghatározása Hornecker módszerével. A megoldás menete az alábbi.

Az $f(x)$ függvényt a $(0,1)$ intervallumon előállitjuk N tagu Taylor részletösszegével

$$P(x)=\sum_{i=0}^{N} a_i x^i$$

majd ezt átirjuk Csebisev polinomok szerinti összegbe

$$P(x)=\sum_{i=0}^{N} b_i T_i^*(x)$$

ebből az összegből kiválasztjuk a pontosságnak megfelelőt

$$P(x)\sim\sum_{i=0}^{M} b_i T_i^*(x)$$

azt perturbáljuk Hornecker módszerével

$$P(x)\sim\sum_{i=0}^{M} b_i^* T_i^*(x)$$

majd azután visszairjuk algebrai polinommá:

$$P(x)\sim\sum_{i=0}^{M} c_i x^i .$$

A FORTRAN lista:

```
      SUBROUTINE PSEMAE(F,N,E,M,H,R)
      DOUBLE PRECISION F,E,H,R,S,T
      DIMENSION F(N),E(N),R(N)
      CALL TAYCSE(F,E,N)
      T=0,0D-0
      IF(H) 1,2,2
1     H=-H
2     DO 3 I=1,N
      S=E(N-I+1)
      IF(S) 4,5,5
4     S=-S
5     T=T+S
      IF(T-H) 3,3,6
3     CONTINUE
      GO TO 7
6     I=N-I+1
      CALL HORNEC(E,I+6,R,H)
      CALL CSETAY(R,E,I)
      M=I
7     RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL PSEMAE(F,N,E,M,H,R)

F,E,R N méretü tömbök,

F tartalma: $a_0, a_1, \ldots, a_{N-1}$ Taylor együtthatók /bemenő/,

E tartalma: $c_0, c_1, \ldots, c_{M-1}$ Taylor együtthatók /kimenő/, e tömbben /eredmény/ az E(M+1),..., E(N) nem tartalmaznak hasznos információt.

N együtthatók száma /bemenő/,

M az eredmény együtthatóinak száma /kimenő/,

H a pontosság /bemenő/,

R munkatömb, tartalma közömbös /be-, kimenő/.

HIVOTT SZUBRUTINOK: TAYCSE,HORNEC,EGYUTT,CSETAY.

b9/ A MINMAX program.

LEIRÁS

Adott $f(x)$ függvény polinommal való legjobb közelitésének meghatározása:

$$P(x) = \sum_{i=0}^{n} a_i x^i$$

a

$$\max_{0 \le x \le 1} |f(x) - P(x)| \quad , \text{vagy} \quad \max_{0 \le x \le 1} \left| 1 - \frac{P(x)}{f(x)} \right|$$

kifejezés minimalizálásával.

A számitás menete a következő. Kiszámitjuk a Csebisev alappontoknak megfelelő bázist. Ez lesz a bázis iterativ javitásához a 0-ik közelités. E bázissal a MAXINT vagy REMINT szubrutinok segitségével meghatározzuk az a_i számok első közelitését. Az eltérés maximumait vizsgálva felveszünk a bázis pontjainak mindkét oldalán egy-egy pontot és e három ponton parabolát fektetünk át, ennek maximumát cseréljük fel a bázis régi pontjával. Igy nyerünk egy uj bázist. Innen folytatható az iterációs ciklus. Az iterációk száma a szubrutin egyik paraméterével beállitható.

A FORTRAN lista:

```
      SUBROUTINE MINMAX(N,E,M,L,S,Z,T,R)
      DOUBLE PRECISION E,W,R,P,S,Z,T,DT
      DIMENSION E(N),W(9),S(N),Z(N),T(N),R(N)
      P=3.141592653589793D-0
      DT=0.5D-2
      W(1)=1.0D-0
      W(2)=0.0D-0
      W(3)=DCOS(P/(N-1))
      W(4)=DSIN(P/(N-1))
      R(1)=W(1)
      CALL FUGGVE(R(1),E(1))
      DO 1 I=2,N
      W(5)=W(1)*W(3)-W(2)*W(4)
      W(2)=W(1)*W(4)+W(2)*W(3)
      W(1)=W(5)
      R(I)=DT*(1+W(1))
      CALL FUGGVE(R(I),E(I))
1     CONTINUE
      L1=1
7     IF(L) 20,20,2
20    CALL REMINT(E,R,N,S,Z)
      GO TO 3
2     CALL MAXINT(E,R,N,S,Z)
3     CONTINUE
      IF(L1-M) 21,21,8
21    N1=N-1
      DO 5 J=2,N1
      W(1)=R(J)-(R(J)-R(J+1))/4
      W(2)=R(J)+(R(J-1)-R(J))/4
      CALL FUGGVE(W(1),W(3))
      CALL FUGGVE(W(2),W(4))
      DO 11 I=7,9
11    W(I)=S(N-1)
      N2=N-2
      DO 10 I=1,N2
      W(7)=S(N-I-1)+W(7)*W(1)
      W(8)=S(N-I-1)+W(8)*W(2)
10    W(9)=S(N-I-1)+W(9)*R(J)
      W(3)=W(3)-W(7)
      W(4)=W(4)-W(8)
      W(5)=(W(4)-W(3))/(W(2)-W(1))
      W(6)=(E(J)-W(9)-W(3))/(R(J)-W(1))
      W(2)=W(2)+W(1)
      W(1)=R(J)+W(1)
      W(1)=DT*(W(1)*W(5)-W(2)*W(6))/(W(5)-W(6))
      Z(J)=W(1)
      CALL FUGGVE(W(1),T(J))
5     CONTINUE
```

```
      L1=L1+1
      DO 6 I=2,N1
      R(I)=Z(I)
6     Z(I)=T(I)
      GO TO 7
8     DO 9 I=1,N
9     E(I)=S(I)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL MINMAX(N,E,M,L,S,Z,T,R)

E,S,Z,T,R N méretü tömbök,

E tartalmazza a közelités együtthatóit: $a_0, a_1, \dots, a_n$ és hibát h /kimenő/,

N=n+2, ahol n a polinom foka /bemenő/,

M az iterációk száma /bemenő/,

L két értéke lehet: L=1 abszolut, L=2 relativ közelités /bemenő/,

S,Z,T,R munkatömbök /be-, kimenő/.

A program számára az $f(x)$ függvény értékeit egy FUGGVE nevü szubrutinnal kell megadni, melynek hivása

CALL FUGGVE(X,Y)

ahol X az $f(x)$ függvény argumentuma és Y az $f(x)$ függvény értéke az X pontban.

HIVOTT SZUBRUTINOK: REMINT,MAXINT,FUGGVE,DSIN,DCOS.

Megjegyzések.

Előfordulhat, hogy az eljárás nem konvergál, ugyanis a kezdeti bázis felvétele az $f(x)$ függvénytől független. A divergencia ugy jelentkezik, hogy a bázis alappontjai /amelyek a $(0,1)$ intervallumban kell, hogy legyenek/ nem lesznek a $(0,1)$ intervallumban. Ez ellen a jelenség ellen ugy lehet védekezni, hogy a számitás egy részére vagy teljes terjedelmére a kvadratikus interpolációt helyettesitjük a MAXKER nevü szubrutinnal /lásd elO/ az alábbi két kártya beszurásával a 10 referenciáju kártya után:

```
CALL MAXKER(W(1),W(3),W(7),R(J),E(J),W(9),W(2),W(4),W(8),Z(J),T(J))
GO TO 5
```

A MAXKER szubrutin a régi bázis alappontját helyettesiti az alappont és két körülötte levő pont közül azzal, amelynél az eltérés maximális abszolut értékü és a régi alappontbeli eltéréssel azonos előjelü.

A kvadratikus interpolácio az $x_0=c$, $x_{n+1}=1$ alappontokat nem változtatja meg. Ha erre szükség volna, akkor e célból egy szubrutint kell irni és hivását beszurni kártyával a MINMAX szubrutinba.

b10/ A MINRAC program.

LEIRÁS

Adott $f(x)$ függvény racionális törttel való legjobb közelitésének meghatározása :

$$R(x)=\frac{\sum_{i=0}^{n} a_i x^i}{\sum_{i=0}^{n} b_i x^i}, \quad \text{vagy} \quad R(x)=\frac{\sum_{i=0}^{n+1} a_i x^i}{\sum_{i=0}^{n} b_i x^i},$$

az eltérés maximumának

$$\max_{0\le x\le 1} \left| f(x)-R(x) \right|$$

minimalizálásával.

A számitás menete a következő. Az $f(x)$ függvényt

$$R^*(x)=\alpha_0+\cfrac{x-x_0}{\alpha_1+\cfrac{x-x_1}{\alpha_2+\dots\cfrac{x-x_{n+1}}{\alpha_n}}}$$

lánctörttel közelitjük /ha n páros födiagonális, ha páratlan födiagonális feletti racionális törtről van szó, lásd fent/. Az x_i alappontoknak a Csebisev alappontokat vesszük.

A λ minimax eltérést

$$\lambda = \min_{\alpha_k} \max_{\{x_k\}} \left| f(x) - R^*(x) \right| ,$$

ugy határozzuk meg /iterációval/, hogy az $\alpha_{n+1} = \infty$ feltétel teljesüljön / $\lambda = 0$ kezdőértékkel indulva ehhez 3 iteráció elég/. Ezután az $f(x) - R^*(x)$ függvény maximum helyeit vizsgálva, kvadratikus interpolációval javitjuk az $\{x_k\}$ alappont sorozatot. Ezután az iterációt többször végrehajthatjuk /az iterációk száma a szubrutin egyik paraméterével beállitható/.

A FORTRAN lista:

```
      SUBROUTINE MINRAC(E,N,M,S,R,Z,P)
      DOUBLE PRECISION E,R,S,Z,P,T,PI,A,B,C,D,F,H,Q,W
      DIMENSION E(N),S(N),R(N),Z(N),P(M)
      PI=3.14159265358979D-0
      L=N-1
      A=1.0D-0
      B=0.0D-0
      C=DCOS(PI/L)
      D=DSIN(PI/L)
      R(1)=A
      CALL FUGGVE(R(1),S(1))
      DO 1 I=2,N
      F=A*C-B*D
      B=A*D+B*C
      A=F
      R(I)=(1+A)/2
      CALL FUGGVE(R(I),S(I))
1     CONTINUE
      N1=N-1
      L1=1
      D=0.0D-0
9     CONTINUE
      DO 2 L=1,3
      T=R(1)
      DO 3 I=1,N
      E(I)=S(I)+T*D
      P(I)=T
3     T=-T
      DO 4 I=2,N1
      A=R(I-1)
      B=E(I-1)
      C=P(I-1)
      DO 4 J=I,N
      F=E(J)-B
      E(J)=(R(J)-A)/F
4     P(J)=-E(J)/F*(P(J)-C)
2     D=D-(E(N-1)-E(N))/(P(N-1)-P(N))
```

```
      IF(L1-M) 20,8,8
2Ø    DO 5 J=2,N1
      A=R(J)+(R(J-1)-R(J))/4
      B=R(J)-(R(J)-R(J+1))/4
      CALL FUGGVE(A,C)
      CALL FUGGVE(B,F)
      Q=E(N-1)
      W=Q
      H=Q
      N2=N-2
      DO 7 I=1,N2
      I1=N-I-1
      H=E(I1)+(A-R(I1))/H
      Q=F(I1)+(B-R(I1))/Q
7     W=E(I1)+(R(J)-R(I1))/W
      C=C-H
      F=F-Q
      H=(F-C)/(B-A)
      Q=(S(J)-W-C)/(R(J)-A)
      A=((A+R(J))*H-(A+B)*Q)/(H-Q)/2
      Z(J)=A
      CALL FUGGVE(A,P(J))
5     CONTINUE
      L1=L1+1
      DO 10 I=2,N1
      R(I)=Z(I)
1Ø    S(I)=P(I)
      GO TO 9
8     DO 11 I=1,N
      P(I)=Ø.ØD-Ø
      S(I)=Ø.ØD-Ø
11    Z(I)=Ø.ØD-Ø
      P(1)=E(N-1)
      S(1)=E(N-2)*P(1)-R(N-2)
      S(2)=1.ØD-Ø
      L1=(N+1)/2
      DO 12 I=2,N2
      A=E(N-I-1)
      B=R(N-I-1)
      IB=(I+3)/2
      Z(1)=S(1)*A-P(1)*B
      DO 13 J=2,IB
13    Z(J)=S(J)*A+P(J-1)-B*P(J)
      DO 14 J=1,IB
      P(J)=S(J)
14    S(J)=Z(J)
12    CONTINUE
      DO 15 I=1,L1
15    E(I)=S(I)/P(1)
      L21=N-L1
      DO 16 I=2,L21
16    E(L1+I)=P(I)/P(1)
      E(L1+1)=0
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL MINRAC(E,N,M,S,R,Z,P)

E,S,R,Z,P N méretü tömbök,

E tartalma: a racionális törtfüggvény együtthatói az alábbi elrendezésben: $K=[(N+1)/2]$-vel

E(1),... E(K) a számláló együtthatói: $a_0, a_1, \dots, a_{k-1}$,

E(K+1) = λ a közelités hibamaximuma,

E(K+2),... E(N) a nevező együtthatói: $b_1, b_2, \dots$, /kimenő/,

a b_0 együttható normálási okokból eggyel egyenlő.

N=2n+2 vagy 2n+3: az alappontok száma /bemenő/,

M az iterációk száma /bemenő/,

S,R,Z,P munkatömbök, értékük közömbös /ki-, bemenő/.

A program számára az $f(x)$ függvény értékeit egy FUGGVE nevü szubrutinnal kell megadni, amelynek hivása

CALL FUGGVE(X,Y)

legyen, ahol X az $f(X)$ függvény argumentuma és Y az függvény értéke az X pontban.

HIVOTT SZUBRUTINOK:DSIN,DCOS,FUGGVE.

Megjegyzések:

Előfordulhat, hogy az eljárás nem konvergál, ugyanis a kezdeti bázis felvétele az $f(x)$ függvénytől független. A divergencia ugy jelentkezik, hogy a bázis alappontjai /amelyek a $(0,1)$ intervallumban kell, hogy legyenek/ nem lesznek a $(0,1)$ intervallumban. Ez ellen a jelenség ellen ugy lehet védekezni, hogy a számitás /egy részére vagy/ teljes terjedelmére a kvadratikus interpolációt helyettesitjük a MAXKER nevü szubrutinnal /lásd el0/ az alábbi két kártya beszurásával a 7 referenciáju kártya után:

```
CALL MAXKER(A,C,H, R(J), S(J),W, B,F,Q,Z(J),P(J))
GO TO 5
```

A MAXKER szubrutin a régi bázis alappontját helyettesiti az alappont és két körülötte levő pont közül azzal, amelynél az eltérés maximális abszolut értékü és a régi alappontbeli eltéréssel azonos előjelü.

A kvadratikus interpoláció az $x_0=0$ és $x_{n+1}=1$ alappontokat nem változtatja meg. Ha erre szükség volna, akkor e célból egy szubrutint kell megirni és hivását beszurni kártyával a MINRAC szubrutinba.

c/ Racionális tört kifejtések.

Racionális tört kifejtések a függvény Taylor sora segitségével nyerhetők Viskovatoff, Wall, Pade, Maehly módszere, valamint a Q.D. algoritmus szerint a FRACS1, FRACJ2, PADE11, PADEMK, CSPAMK szubrutinok felhasználásával.

c1/ A FRACS1 program.

LEIRÁS

A

$$P(x) = c_0 + c_1 x + \dots + c_{N-1} x^{N-1}$$

polinomot az alábbi tipusu lánctörtbe irja át:

$$L(x) = \cfrac{\alpha_1}{\alpha_0 + \cfrac{\alpha_2 x}{\alpha_1 + \cfrac{\alpha_3 x}{\alpha_2 + \ddots + \cfrac{\alpha_{N-1} x}{\alpha_{N-2}}}}}.$$

A számitás menete: legyen

$$\beta_{1,n} = c_n, \quad \beta_{2,n} = -c_{n+1}, \quad n = 0,1,2,\dots$$

valamint

$$\beta_{m,n} = \beta_{m-1,0} \cdot \beta_{m-2,n+1} - \beta_{m-2,0} \cdot \beta_{m-1,n+1},$$

$m=2,3,\dots,\ ;\ n=0,1,2,\dots,$

akkor

$$\beta_{m,0}=\alpha_m \qquad m=2,3,\dots .$$

A FORTRAN lista:

```
      SUBROUTINE FRACS1(A,N,E)
      DOUBLE PRECISION A,E,B,C
      DIMENSION A(N),E(N)
      A(1)=1.0D-2
      DO 1 I=2,N
1     E(I)=-A(I)
      DO 3 I=2,N
3     A(N-I+2)=A(N-I+1)
      N1=N-2
      DO 2 K=2,N1
      B=A(K)
      C=E(K)
      IB=N1-K+2
      DO 2 I=1,IB
      NB=N-I+1
      E(NB)=C*A(NB)-B*E(NB)
2     A(NB)=E(NB-1)
      A(N)=E(N)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL FRACS1(A,N,E)

A,E N méretü tömbök,

A tartalma: $a_0, a_1, \dots, a_{N-1}$ Taylor együtthatók /bemenő/,

E tartalma: $\alpha_0, \alpha_1, \dots, \alpha_{N-1}$ lánctört együtthatók /kimenő/,

N az együtthatók száma /bemenő/.

Megjegyzés.

Bár a módszer tetszőleges $\{a_i\}$ sorozatra alkalmazható, mégis a program hibás eredményt ad, ha valamelyik α_k szám nem ábrázolható a gépen /ha $\{a_i\}$ gyorsan nő vagy csökken, akkor a $\beta_{m,n}$ sorozat hatványozottabban nő vagy csökken/.

c2/ A FRACJ2 program.

LEIRÁS

A

$$P(x)=f_0+f_1x+\dots+f_{N-1}x^{N-1}$$

polinomot átirja a

$$W(x)=\cfrac{c_0}{1+b_0x-\cfrac{c_1x^2}{1+b_1x-\cfrac{c_2x^2}{1+b_2x-\dots}}}$$

lánctörtbe (Wall [2]).

A FORTRAN lista:

```
      SUBROUTINE FRACJ2(C,A,B,N1,N)
      DOUBLE PRECISION C,A,B,S,Z,P,Q,R
      DIMENSION C(N1),A(N),B(N)
      COMMON/MUNK/R(612)
      N3=3*N+3
      DO 1 I=1,N3
1     R(I)=0.0D-0
      R(2*N+4)=1.0
      A(1)=C(1)
      B(1)=-C(2)/C(1)
      S=A(1)
      Z=B(1)
      N2=N-1
      DO 2 K=1,N2
      K1=K+1
      DO 3 I=1,K1
      R(I)=R(N+I+1)
3     R(N+I+1)=R(2*N+I+2)
      DO 4 I=1,K1
4     R(2*N+I+3)=B(K)*R(N+I+2)+R(N+I+1)-A(K)*R(I+1)
      P=0.0D-0
      Q=0.0D-0
      DO 5 I=1,K1
      P=P+R(2*N+I+3)*C(I+K)
5     Q=Q+R(2*N+I+3)*C(I+K+1)
      A(K+1)=P/S
      S=P
      B(K+1)=-Q/S-Z
2     Z=-Q/S
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL FRACJ2 (C,A,B,N1,N)

C N1 méretü tömb, N1=2*N

A,B N méretü tömbök,

C tartalma: $f_0, f_1, \ldots, f_{2N-1}$ Taylor együtthatók /bemenő/,

A tartalma: $c_0, c_1, \ldots, c_{N-1}$ /kimenő/,

B tartalma: $b_0, b_1, \ldots, b_{N-1}$ /kimenő/,

N tartalma: az együtthatók száma/bemenő/.

c3/ A PADE11 program.

LEIRÁS

A

$$P(x) = a_0 + a_1 x + \cdots + a_{N-1} x^{N-1}$$

polinomot a

$$Q(x) = \cfrac{c_0}{1 - \cfrac{c_1 x}{1 - \cfrac{c_2 x}{1 - \ddots - c_{N-1} x}}},$$

alaku lánctörtté irja át.

A számitás a Q.D. algoritmus alapján megy:

$$q_1^k = \frac{a_{k+1}}{a_k}, \quad k = 0,1,2,\ldots$$

$$e_1^k = q_1^{k+1} - q_1^k, \quad k = 0,1,2,\ldots$$

$$\left.\begin{aligned} q_{r+1}^k &= q_r^{k+1} \frac{e_r^{k+1}}{e_r^k} \\ e_r^k &= e_{r-1}^{k+1} + q_r^{k+1} - q_r^k \end{aligned}\right\} \quad \begin{aligned} &k = 0,1,2,\ldots, \\ &r = 1,2,\ldots . \end{aligned}$$

Az algoritmus használható

$$Q^*(x) = a_0 + a_1x + \dots + a_kx^k + \cfrac{a_{k+1}x^{k+1}}{1 - \cfrac{q_1^k x}{1 - \cfrac{e_1^k x}{1 - \dots}}}$$

tipusu előállitás készitésére is.

A FORTRAN lista:

```
      SUBROUTINE PADE11(F,E,N,R,S)
      DOUBLE PRECISION F,E,R,S
      DIMENSION F(N),E(N),R(N),S(N)
      E(1)=F(1)
      E(2)=F(2)/F(1)
      DO 1 I=3,N
      E(I)=F(I)/F(I-1)
    1 R(I)=E(I)-E(I-1)
      NL=1
      DO 2 J=4,N
      IF(NL) 3,8,8
    8 DO 4 K=J,N
    4 S(K)=R(K)/R(K-1)*E(K-1)
      GO TO 5
    3 DO 6 K=J,N
    6 S(K)=R(K)-R(K-1)+E(K-1)
    5 E(J-1)=R(J-1)
      DO 7 K=J,N
      E(K)=R(K)
    7 R(K)=S(K)
    2 NL=-NL
      E(N)=R(N)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL PADE11 (F,E,N,R,S)

F,E,R,S N méretü tömbök,

F tartalma: $a_0, a_1, \dots, a_{N-1}$ Taylor együtthatók /bemenő/,

E tartalma: $c_0, c_1, \dots, c_{N-1}$ a Q.D. tört együtthatók /kimenő/,

N az együtthatók száma /bemenő/,

R,S munkatömbök, tartalmuk közönbös /be-, kimenő/.

c4/ A PADEMK program.

LEIRÁS

Az $f(x)$ függvény Padé közelitése az a

$$R_{m,k}(x)=\frac{\sum_{i=0}^{m} p_i x^i}{1+\sum_{i=1}^{k} q_i x^i}$$

racionális tört, amelyre $x \to 0$ esetén az

$$f(x)-R_{m,k}(x)=O(x^{m+k+1})$$

reláció érvényes. Az ordo tag együtthatóját $d_{m,k}$ -t jelöljük λ -val. A λ értéke a Padé közelités pontosságát jellemzi.

A p_i és q_i számok kiszámitása az alábbi képletek szerint történik. Először a q_i számok határozhatók meg a

$$\sum_{l=1}^{n^{**}} a_{p-l}\, q_l = -a_p \,,$$

$$p=m+1, m+2, \ldots, m+k \,; \qquad n^{**}=\min(p,k),$$

egyenletrendszerből. A p_i értékekre viszont

$$p_i=\sum_{l=0}^{n^*} a_{i-l}\, q_l \,, \qquad n^*=\min(i,k)$$

érvényes. Az a_i számok $f(x)$ Taylor sorának együtthatói.

A FORTRAN lista:

```
      SUBROUTINE PADEMK(F,N,E1,M,E2,K,EPS,A,B)
      DOUBLE PRECISION F,E1,E2,S,A,B,EPS
      DIMENSION F(N),E1(M),E2(K),A(K,K),B(K,K)
      K1=K-1
      DO 1 I=1,K1
      K2=M+I-1
      IF(K2-K1) 9,9,2
2     K2=K1
9     DO 1 J=1,K2
1     A(I,J)=F(M+I-J)
      DO 3 I=1,K1
3     A(I,K)=-F(M+I)
```

```
      CALL MEGOLD(A,B,K)
      E2(1)=1.0D-0
      DO 4  I=1,K1
4     E2(I+1)=A(K,I)
      E1(1)=F(1)
      M1=M-1
      DO 5 I=1,M1
      S=F(I+1)
      K2=I
      IF(K2-K1) 8,8,6
6     K2=K1
8     DO 7 J=1,K2
7     S=S+F(I+1-J)*E2(J+1)
5     E1(I+1)=S
      EPS=F(N)
      DO 10 I=1,K1
10    EPS=EPS+F(N-I)*E2(I+1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL PADEMK (F,N,E1,M,E2,K,EPS,A,B)

F N méretü tömb, tartalma $a_0, a_1, \ldots, a_{N-1}$ Taylor együtthatók /bemenő/,

N az együtthatók száma N=m+k+2 /bemenő/,

E1 M méretü tömb, tartalma $p_0, p_1, \ldots, p_{M-1}$ együtthatók /kimenő/,

M együtthatók száma M=m+1 /bemenő/,

E2 K méretü tömb, tartalma $q_1, q_2, \ldots, q_k$ együtthatók /kimenő/,

K együtthatók száma K=k+1 /bemenő/,

EPS $= \lambda = \sum_{j=0}^{k} a_{N-j} \cdot q_j$ hibatag /kimenő/,

A,B kétdimenziós K méretü munkatömbök /be-, kimenő/.

HIVOTT SZUBRUTINOK: MEGOLD.

c5/ A CSPAMK program.

LEIRÁS

Az $f(x)$ függvény Csebisev - Pade közelitésének nevezzük a

$$CP(x)=\frac{\sum_{i=0}^{m} a_i p_i(x)}{1+\sum_{i=1}^{k} b_i p_i(x)}$$

tört kifejezést, amelyben a $p_i(x)$ polinomok kielégitik a

$$p_k(x)\,p_l(x)=\frac{1}{2}\left[p_{k+l}(x)+p_{|k-l|}(x)\right]$$

relációt. Az a_i és b_i számok a

$$\left[\sum_{i=0}^{\infty} c_i p_i(x)\right]\left[1+\sum_{i=1}^{k} b_i p_i(x)\right]-\left[\sum_{i=0}^{m} a_i p_i(x)\right]=\lambda\, p_{m+k+1}(x)+\dots,$$

azonosságból számithatók ki. Feltételezzük, hogy az függvény az

$$f(x)=\sum_{i=0}^{\infty} c_i p_i(x)$$

sorral állitható elő és a c_i számok adottak.

A számitás az alábbi képletek szerint történik: $b_j\ \ j=1,2,\dots k$

$$\frac{1}{2}\sum_{j=0}^{k}\left(c_{l+j}+c_{|l-j|}\right)b_j=-c_l\ ,\quad l=m+1,\ m+2,\dots,\ m+k\ ,$$

egyenletrendszerből nyerhető, b_j ismeretében pedig:

$$a_0=\frac{1}{2}\sum_{j=0}^{k} a_j b_j\ ,$$

$$a_l=\frac{1}{2}\sum_{j=0}^{k}\left(c_{l+j}+c_{|l-j|}\right)b_j\ ;\quad j=1,2,\dots m,$$

$$\lambda=\frac{1}{2}\sum_{j=0}^{k}\left(c_{m+k+j+1}+c_{m+k-j+1}\right)b_j\ ,\quad b_0=1\ .$$

A FORTRAN lista:

```
      SUBROUTINE CSPAMK(F,N,E1,M,E2,K,EPS,A,B)
      DOUBLE PRECISION F,E1,E2,EPS,S,A,B
      DIMENSION F(N),E1(M),E2(K),A(K,K),B(K,K)
      K1=K-1
      DO 1 I=1,K1
      A(I,K)=-F(M+I)
      DO 1 J=1,K1
      I1=I-J+M-1
      IF(I1) 8,9,9
 8    I1=-I1
 9    CONTINUE
 1    A(I,J)=(F(M+I+J)+F(I1+1))/2
      CALL MEGOLD(A,B,K)
      E2(1)=1,0D-0
      DO 2 I=2,K
 2    E2(I)=A(K,I-1)
      S=F(1)
      DO 3 I=2,K
 3    S=S+F(I)*E2(I)
      E1(1)=S/2
      DO 4 I=2,M
      S=F(I)*2
      DO 5 J=2,K
      K2=I-J
      IF(K2) 6,10,10
 6    K2=-K2
 10   CONTINUE
 5    S=S+(F(J+I-1)+F(K2+1))*E2(J)
 4    E1(I)=S/2
      K2=M+K
      S=F(K2)*2
      DO 7 I=1,K1
 7    S=S+(F(K2+I)+F(K2-I))*E2(I+1)
      EPS=S/2
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL CSPAMK $\left(\text{F,N,E1,M,E2,K,EPS,A,B}\right)$

F N méretü tömb, tartalma a $2c_0, c_1, \ldots, c_{N-1}$ együtthatók /bemenő/,

N az együtthatók száma N=m+2K /bemenő/,

E1 M méretü tömb, tartalma az $a_0, a_1, \dots, a_{M-1}$ együtthatók /kimenő/,

M az együtthatók száma M=m+1 /bemenő/,

E2 K méretü tömb, tartalma a $b_1, b_2, \dots, b_{K-1}$ együtthatók /kimenő/,

K az együtthatók száma K=k+1, /bemenő/,

EPS= λ /kimenő/,

A,B kétdimenziós K méretü munkatömbök /be-, kimenő/,

HIVOTT SZUBRUTINOK: MEGOLD.

Megjegyzés:

A végeredményt a RENDZ3 szubrutin tudja közönséges polinomok hányadosává visszarendezni, ha $p_i(x) = T_i(x)$ vagy $p_i(x) = T_i^*(x)$ Csebisev polinom.

d/ Hipergeometriai tipusu függvények Csebisev sorfejtése.

A Taylor sor részletösszege helyett a numerikus stabilitás megóvása miatt sokszor célszerü a Csebisev sor részletösszegével dolgozni. A Csebisev sor együtthatóinak kiszámitására direkt módszereket alkalmaznak a HIPE32,HIPE43,SHIP32,SHIP43 szubrutinok. /[4]/.

d1/ A HIPE32 program.

LEIRÁS

Az

$$f(x) = {}_3F_2(a_1, a_2, a_3; b_1, b_2; -\lambda x) = \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k (a_3)_k}{(b_1)_k (b_2)_k \, k!} (-\lambda x)^k, \quad 0 \le x \le 1,$$

függvény

$$f(x) = \sum_{k=0}^{\infty} c_k T_k^*(x)$$

Csebisev sorának c_k együtthatóit számitja ki.

A számitás rekurziós képletet használ:

$$\frac{1}{2}\left(Z_1 c_{n+1} + Z_2 c_{n+2} + Z_3 c_{n+3}\right) = \qquad n=0,1,2,\ldots$$

$$= S_0 c_n + S_1 c_{n+1} + S_2 c_{n+2} + S_3 c_{n+3} + S_4 c_{n+4},$$

$$Z_1 = n + b_1 + b_2 - 3/2 + \frac{l_{3/2}}{n+3/2},$$

$$Z_2 = -2n - 4 + \frac{l_{3/2}}{n+3/2} + \frac{l_{3/2}}{n+5/2},$$

$$Z_3 = n - b_1 - b_2 + \frac{11}{2} + \frac{l_{3/2}}{n+5/2},$$

$$l_{3/2} = (b_1 - 3/2)(b_2 - 3/2),$$

$$S_0 = n + a_1 + a_2 + a_3 - \frac{5}{2} + \frac{2k_1}{n+1} - \frac{2k_{3/2}}{n+3/2},$$

$$S_1 = -4n - 2(a_1 + a_2 + a_3) + 1 + \frac{2k_{3/2}}{n+3/2} - \frac{2k_{3/2}}{n+5/2},$$

$$S_2 = 6n + 12 - \frac{2k_1}{n+1} + \frac{2k_{3/2}}{n+3/2} + \frac{2k_{3/2}}{n+5/2} - \frac{2k_1}{n+3},$$

$$S_3 = -4n + 2(a_1 + a_2 + a_3) - 17 - \frac{2k_{3/2}}{n+3/2} + \frac{2k_{3/2}}{n+5/2},$$

$$S_4 = n - (a_1 + a_2 + a_3) + \frac{13}{2} - \frac{2k_{3/2}}{n+5/2} + \frac{2k_1}{n+3},$$

$$k_a = (a_1 - a)(a_2 - a)(a_3 - a).$$

A rekurziós képlet megoldása csökkenő $n = N, N-1, \ldots, 1, 0$ mentén történik, normalizációval.

A FORTRAN lista:

```
      SUBROUTINE HIPE32(A1,A2,A3,B1,B2,RM,K,N,E,L,FK)
      DOUBLE PRECISION A1,A2,A3,B1,B2,RM,E,FK,R,S
      DIMENSION E(N),S(25)
      COMMON/MUNK/R(612)
      DO 1 I=1,5
1     S(I)=(1+I)/(2.ØD-Ø)
      S(6)=(S(1)-S(2)/B1)*(S(1)-S(2)/B2)
      DO 2 I=1,2
2     S(I+7)=(S(1)-S(I)/A1)*(S(1)-S(I)/A2)*(S(1)-S(I)/A3)*S(3)
      S(11)=B1+B2
      S(12)=A1+A2+A3
      S(13)=4*B1*B2/(A1*A2*A3*RM)
      S(24)=(S(1)/B1)/B2
      S(25)=((S(1)/A1)/A2)/A3
      DO 3 L1=1,L
      S(7)=K-1
      M=3Ø4*(L1-1)
      KM=K+M+1
      R(KM)=1.ØD-7Ø
      IF(L1-1) 24,4,24
24    R(KM+1)=1.ØD-69
      GO TO 5
4     R(KM+1)=Ø.ØD-Ø
5     R(KM+2)=Ø.ØD-Ø
      R(KM+3)=Ø.ØD-Ø
      DO 6 I=1,K
      S(16)=S(6)/(S(7)+S(4))
      S(15)=S(6)/(S(7)+S(2))
      S(14)=S(15)+(S(7)+S(11)-S(2))*S(24)
      S(15)=S(15)+S(16)-(2*S(7)+4)*S(24)
      S(16)=S(16)+(S(7)-S(11)+3.5)*S(24)
      S(2Ø)=S(8)/(S(7)+S(1))
      S(21)=S(9)/(S(7)+S(2))
      S(22)=S(9)/(S(7)+S(4))
      S(23)=S(8)/(S(7)+S(5))
      S(17)=S(2Ø)-S(21)+(S(7)+S(12)-S(4))*S(25)
      S(18)=S(21)-S(22)-(4*S(7)+2*S(12)-1)*S(25)
      S(19)=S(21)-S(2Ø)+S(22)-S(23)+(6*S(7)+12)*S(25)
      S(2Ø)=S(22)-S(21)-(4*S(7)-2*S(12)+17)*S(25)
      S(21)=S(23)-S(22)+(S(7)-S(12)+6.5)*S(25)
      J=K+M+1-I
      R(J)=(S(13)*S(14)-S(18))*R(J+1)+(S(13)*S(15)-S(19))*R(J+2)
      R(J)=(R(J)+(S(13)*S(16)-S(2Ø))*R(J+3)-S(21)*R(J+4))/S(17)
6     S(7)=S(7)-S(1)
      R(M+1)=R(M+1)/S(3)
      S(10)=R(M+1)
```

```
      DO 7 I=2,K
7     S(1Ø)=S(1Ø)+R(M+I)
      S(22)=S(1)
      DO 8 I=1,K
      R(M+I)=R(M+I)*S(22)/S(1Ø)
8     S(22)=-S(22)
3     CONTINUE
      IF(L-1) 29,29,9
29    DO 1Ø I=1,N
1Ø    E(I)=R(I)
      GO TO 13
9     S(1Ø)=S(1)
      DO 11 I=22,25
11    S(I)=Ø.ØD-Ø
      DO 12 I=1,K
      S(22)=S(22)+R(I)
      S(23)=S(23)+R(I)*S(1Ø)
      S(24)=S(24)+R(I+3Ø4)
      S(25)=S(25)+R(I+3Ø4)*S(1Ø)
12    S(1Ø)=-S(1Ø)
      S(1Ø)=S(22)*S(25)-S(23)*S(24)
      S(24)=(FK*S(25)-S(24))/S(1Ø)
      S(25)=(S(22)-FK*S(23))/S(1Ø)
      DO 14 I=1,N
14    E(I)=S(24)*R(I)+S(25)*R(I+3Ø4)
13    RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL HIPE32(A1,A2,A3,B1,B2,RM,K,N,E,L,FK)

$\left.\begin{array}{l} \text{A1} \\ \text{A2} \\ \text{A3} \end{array}\right\}$ számláló paraméterek $\neq 0,-1,-2,\ldots$ /bemenő/,

$\left.\begin{array}{l} \text{B1} \\ \text{B2} \end{array}\right\}$ nevező paraméterek $\neq 0,-1,-2,\ldots$ /bemenő/,

RM a függvény argumentuma $= \lambda$ /bemenő/,

K az iteráció kezdő értéke /bemenő/,

N az együtthatók száma /bemenő/,

E N méretü tömb, tartalma: $c_0, c_1, \dots, c_{N-1}$ együtthatók /kimenő/,
L szükséges sorozatok száma L=1 vagy L=2 /bemenő/,
FK kapcsoló érték = $f(x)\big|_{x=1}$, ha L=2, különben tetszőleges /bemenő/.

Megjegyzések.

A nyert sorozat pontosságáról ugy nyerhetünk információt, hogy elvégeztetjük a számitást egy konkrét K értékkel, majd pl: a K'=K+10 értékkel és a két sorozatot összehasonlitjuk. A K értéke L=1 esetén 612, L=2 esetén 306 alatt kell legyen.

A szubrutin ${}_3F_2$ függvényen kivül alkalmas az ${}_3F_1, {}_3F_0$, ${}_2F_2, {}_2F_1, {}_2F_0, {}_1F_2, {}_1F_1, {}_1F_0, {}_0F_2, {}_0F_1$ függvények Csebisev együtthatóinak kiszámitására a konfluencia elv alapján. Ez abból áll, hogy az ${}_3F_2$ paramétereinek és argumentumának alkalmas végtelenhez tartása esetén egy másik hipergeometriai függvényhez konvergál, pl.:

$$\lim_{\substack{b_1\to\infty \\ b_2\to\infty}} {}_3F_2(a_1,a_2,a_3;b_1,b_2;-\lambda b_1 b_2 x) = {}_3F_0(a_1,a_2,a_3;;-\lambda x).$$

A szubrutinban az $a_1 = 10^{25}$ helyettesités megvalósitja az $a_1 \to \infty$ határátmenetet.

Az ${}_3F_1, {}_3F_0, {}_2F_0$ függvények esetében L=2 veendő, ekkor a szubrutin automatikusan két sorozatot számol és abból lineár kombinációként állitja elő a végső $c_0, c_1, \dots, c_{N-1}$ sorozatot.

A K értékét ajánlatos a függvény analitikai tulajdonságainak megfelelően választani. Ha a függvény sora gyorsan konvergál, elégséges a K=N+10 választás. Lassu konvergencia esetén /pl: az előző három esetben/ célszerü a K=2*N választás.

d2/ A HIPE43 program.

LEIRÁS

A program az

$$f(x) = {}_4F_3(a_1, a_2, a_3, a_4; b_1, b_2, b_3; -\lambda x) =$$

$$= \sum_{k=0}^{\infty} \frac{(a_1)_k (a_2)_k (a_3)_k (a_4)_k}{(b_1)_k (b_2)_k (b_3)_k \, k!} (-\lambda x)^k, \quad 0 \le x \le 1.$$

függvény Csebisev sorának

$$f(x) = \sum_{k=0}^{\infty} c_k T_k^*(x)$$

együtthatóit számitja ki.

A számitás rekurziós képletet használ:

$$\frac{4}{\lambda}\left(U_1 c_{n+1} + U_2 c_{n+2} + U_3 c_{n+3} + U_4 c_{n+4}\right) =$$

$$= V_0 c_n + V_1 c_{n+1} + V_2 c_{n+2} + V_3 c_{n+3} + V_4 c_{n+4} + V_5 c_{n+5},$$

$$n = 0, 1, 2, 3, \dots,$$

$$U_1 = n + p - 7/2 + \frac{l_{3/2}}{n+3/2} - \frac{l_2}{n+2},$$

$$U_2 = -3n - p - 3/2 + \frac{l_{3/2}}{n+3/2} - \frac{l_2}{n+3},$$

$$U_3 = 3n - p + \frac{27}{2} + \frac{l_2}{n+2} - \frac{l_{3/2}}{n+7/2},$$

$$U_4 = -n + p - \frac{17}{2} + \frac{l_2}{n+3} - \frac{l_{3/2}}{n+7/2},$$

$$p = b_1 + b_2 + b_3, \qquad l_a = 2(b_1 - a)(b_2 - a)(b_3 - a),$$

$$V_0 = n + q - 9/2 + \frac{k_1}{n+1} - \frac{2k_{3/2}}{n+3/2} + \frac{k_2}{n+2},$$

$$V_1 = -5n - 3q + \frac{17}{2} + \frac{2k_{3/2}}{n+3/2} - \frac{k_1+2k_2}{n+2} + \frac{k_2}{n+3},$$

$$V_2 = 10n + 2q + 11 - \frac{k_1}{n+1} + \frac{2k_{3/2}}{n+3/2} - \frac{k_1+2k_2}{n+3} + \frac{2k_{3/2}}{n+7/2},$$

$$V_3 = -10n + 2q - 39 + \frac{k_1+2k_2}{n+2} - \frac{2k_{3/2}}{n+3/2} - \frac{2k_{3/2}}{n+7/2} + \frac{k_1}{n+4},$$

$$V_4 = 5n - 3q + \frac{67}{2} - \frac{k_2}{n+2} + \frac{k_1+2k_2}{n+3} - \frac{2k_{3/2}}{n+7/2},$$

$$V_5 = -n + q - \frac{19}{2} - \frac{k_2}{n+3} + \frac{2k_{3/2}}{n+7/2} - \frac{k_1}{n+4},$$

$$q = a_1 + a_2 + a_3 + a_4,$$

$$k_a = 2(a_1-a)(a_2-a)(a_3-a)(a_4-a).$$

A rekurziós képlet megoldása csökkenő n=N,N-1,... 1,0 mentén történik, normalizációval.

A FORTRAN lista:

```
      SUBROUTINE HIPE43(A1,A2,A3,A4,B1,B2,B3,RM,K,N,E,L,FK)
      DOUBLE PRECISION A1,A2,A3,A4,B1,B2,B3,RM,R,E,S,FK
      DIMENSION E(N),S(34)
      COMMON/MUNK/R(612)
      DO 1 I=1,8
1     S(I)=(1,0D-0+I)/2,0D-0
      DO 3 I=2,3
3     S(I+7)=S(3)*(S(1)-S(I)/B1)*(S(1)-S(I)/B2)*(S(1)-S(I)/B3)
```

```
      DO 2 I=1,3
      S(I+11)=(S(1)-S(I)/A1)*(S(1)-S(I)/A2)*(S(1)-S(I)/A3)
2     S(I+11)=S(I+11)*(S(1)-S(I)/A4)*S(3)
      S(15)=A1+A2+A3+A4
      S(16)=B1+B2+B3
      S(13)=S(3)*S(13)
      S(17)=S(7)/RM
      DO 7 L1=1,L
      S(11)=K-1
      LB=305*(L1-1)
      R(K+LB+1)=1,0D-70
      IF(L1-1) 20,4,20
20    R(K+LB+2)=1,0D-69
      GO TO 5
4     R(K+LB+2)=0,0D-0
5     DO 6 I=3,5
6     R(K+LB+I)=0,0D-0
      S(33)=((S(1)/B1)/B2)/B3
      S(34)=(((S(1)/A1)/A2)/A3)/A4
      DO 8 I=1,K
      S(22)=S(10)/(S(11)+S(3))
      S(23)=S(10)/(S(11)+S(5))
      S(24)=S(9)/(S(11)+S(2))
      S(25)=S(9)/(S(11)+S(6))
      S(18)=S(24)-S(22)+(S(11)+S(16)-S(6))*S(33)
      S(19)=S(24)-S(23)-(S(11)*3+S(16)+S(2))*S(33)
      S(20)=S(22)-S(25)+(S(11)*3-S(16)+S(6)+10)*S(33)
      S(21)=S(23)-S(25)-(S(11)-S(16)+S(6)+5)*S(33)
      S(24)=S(12)/(S(11)+S(1))
      S(25)=S(12)/(S(11)+S(3))
      S(26)=S(13)/(S(11)+S(2))
      S(27)=S(14)/(S(11)+S(3))
      S(28)=S(12)/(S(11)+S(5))
      S(29)=S(14)/(S(11)+S(5))
      S(30)=S(13)/(S(11)+S(6))
      S(31)=S(12)/(S(11)+S(7))
      S(22)=S(24)-S(26)+S(27)+(S(11)+S(15)-S(8))*S(34)
      S(23)=S(26)-S(25)-S(27)*2+S(29)-(S(11)*5+S(15)*3-8,5)*S(34)
      S(24)=S(26)-S(24)-S(29)*2-S(28)+S(30)+(S(11)*10+S(15)*2+11)*S(34)
      S(25)=S(27)*2+S(25)-S(26)-S(30)+S(31)-(S(11)*10-S(15)*2+39)*S(34)
      S(26)=S(28)+S(29)*2-S(27)-S(30)+(S(11)*5-S(15)*3+33,5)*S(34)
      S(27)=S(30)-S(29)-S(31)-(S(11)-S(15)+9,5)*S(34)
      K1=K+LB+1-I
      R(K1)=(S(17)*S(18)-S(23))*R(K1+1)+(S(17)*S(19)-S(24))*R(K1+2)
      R(K1)=R(K1)+(S(17)*S(20)-S(25))*R(K1+3)
      R(K1)=(R(K1)+(S(17)*S(21)-S(26))*R(K1+4)-S(27)*R(K1+5))/S(22)
8     S(11)=S(11)-S(1)
      R(LB+1)=R(LB+1)/S(3)
      S(38)=R(LB+1)
      DO 10 I=2,K
10    S(38)=S(38)+R(LB+I)
```

```
      S(29)=S(1)
      DO 11 I=1,K
      R(I+LB)=R(I+LB)*S(29)/S(30)
11    S(29)=-S(29)
7     CONTINUE
      IF(L-1) 21,21,9
21    DO 16 I=1,N
16    E(I)=R(I)
      GO TO 15
9     S(28)=S(1)
      DO 12 I=29,32
12    S(I)=0.0D-0
      DO 13 I=1,K
      S(29)=S(29)+R(I)
      S(30)=S(30)+R(I)*S(28)
      S(31)=S(31)+R(305+I)
      S(32)=S(32)+R(305+I)*S(28)
13    S(28)=-S(28)
      S(28)=S(29)*S(32)-S(30)*S(31)
      S(31)=(FK*S(32)-S(31))/S(28)
      S(32)=(S(29)-FK*S(30))/S(28)
      DO 14 I=1,N
14    E(I)=S(31)*R(I)+S(32)*R(305+I)
15    RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL HIPE43(A1,A2,A3,A4,B1,B2,B3,RM,K,N,E,L,FK)

A1,A2,A3,A4 számláló paraméterek $\neq$ 0,-1,-2,... /bemenő/,

B1,B2,B3 nevező paraméterek $\neq$ 0,-1,-2,... /bemenő/,

RM a függvény argumentuma= λ , lásd a megjegyzéseket /bemenő/,

K az iteráció kezdő értéke /bemenő/,

N az együtthatók száma /bemenő/,

E N méretü tömb, tartalma: $c_0, c_1, \ldots, c_{N-1}$ együtthatók /kimenő/,

L szükséges sorozatok száma: L=1, vagy L=2 /bemenő/,

FK kapcsoló érték = $f(x)\big|_{x=1}$, ha L=2, különben közönbös /bemenő/.

Megjegyzések:

A nyert sorozat pontosságáról ugy nyerhetünk információt, hogy elvégeztetjük a számitást egy konkrét K értékkel, majd pl.: a K'=K+10 értékkel és a két eredményt összehasonlitjuk. A K értéke L=1 esetén 612, L=2 esetén 306 alatt legyen.

A szubrutin az ${}_4F_3$ függvényen kivül alkalmas az ${}_4F_2$, ${}_4F_1$, ${}_4F_0$, ${}_3F_3$, ${}_3F_2$, ${}_3F_1$, ${}_3F_0$, ${}_2F_3$, ${}_2F_2$, ${}_2F_1$, ${}_2F_0$, ${}_1F_3$, ${}_1F_2$, ${}_1F_1$, ${}_1F_0$, ${}_0F_3$, ${}_0F_2$, ${}_0F_1$, függvények Csebisev együtthatóinak számitására is a konfluencia elv alapján. Ez abban áll, hogy az függvény paramétereinek és argumentumának megfelelő végtelenhez tartása esetén egy másik hipergeometriai függvényhez konvergál, pl.:

$$\lim_{\substack{b_1\to\infty\\ b_2\to\infty\\ b_3\to\infty}} {}_4F_3(a_1,a_2,a_3,a_4;\ b_1,b_2,b_3;\ -\lambda b_1 b_2 b_3 x) = {}_4F_0(a_1,a_2,a_3,a_4;;\ -\lambda x).$$

A szubrutinban az $a_1 = 10^{25}$ helyettesités megvalósitja az $a_1 \to \infty$ határátmenetet.

A szubrutin ugy van felépitve, hogy a többszörös konfluencia is könnyen megvalósitható legyen; ezért RM-t mint módositott argumentumot az alábbi formában kell megadni

$$RM = \lambda \frac{a_1 a_2 a_3 a_4}{b_1 b_2 b_3}.$$

Az ${}_4F_0$, ${}_4F_1$, ${}_4F_2$, ${}_3F_0$, ${}_3F_1$, ${}_2F_0$ függvények esetében L=2 veendő, ekkor a program automatikusan két sorozatot számol és abból lineár kombinációként állitja elő a végső $c_0, c_1, \dots c_{N-1}$ sorozatot.

A K értékét ajánlatos a függvény analitikai tulajdonságainak megfelelően választani. Durván, ha a függvény sora gyorsan konvergál, elégséges a K=N+10 választás. Lassu konvergencia esetén /pl: a fenti öt esetben/ célszerü a K=2*N választás.

d3/ A SHIP32 program.

LEIRÁS

Az

$$f_1(x) = {}_3F_2(a_1, a_2, a_3; b_1, b_2; \lambda x)$$

Csebisev sorának

$$f_1(x) = \sum_{k=0}^{\infty} c_k T_k(x), \quad -1 \le x \le 1,$$

együtthatóit, valamint az

$$f_2(x) = \mathrm{Re}\, f_1(ix), \quad f_3(x) = \mathrm{Im}\, f_1(ix),$$

függvények

$$f_2(x) = \sum_{k=0}^{\infty} d_{2k}(-1)^k T_{2k}(x),$$

$$f_3(x) = 2\sum_{k=0}^{\infty} d_{2k+1}(-1)^k T_{2k+1}(x). \qquad -1 \le x \le 1$$

Csebisev sorainak együtthatóit határozza meg.

A számitás rekurziós képletet használ:

$$\frac{2}{\lambda}\left(P_1 c_{k+1} + SG.P_2 c_{k+3} + P_3 c_{k+5}\right) =$$

$$= R_0 c_k + SG.R_1 c_{k+2} + R_2 c_{k+4} + SG.R_3 c_{k+6},$$

$$k = 0, 1, 2, \ldots,$$

$$P_1 = k + b_1 + b_2 - 2 + \frac{l_2}{k+2},$$

$$P_2 = 2k + 6 - \frac{l_2}{k+2} - \frac{l_2}{k+4}, \qquad l_2 = (b_1 - 2)(b_2 - 2),$$

$$P_3 = k - b_1 - b_2 + 8 + \frac{l_2}{k+4},$$

$$R_0 = k + q - 3 + \frac{k_1}{k+1} - \frac{k_2}{k+2},$$

$$R_1 = 3k + q + 3 - \frac{k_1}{k+1} + \frac{k_2}{k+4},$$

$$R_2 = 3k - q + 15 + \frac{k_2}{k+2} - \frac{k_1}{k+5},$$

$$R_3 = k - q + 9 - \frac{k_2}{k+4} + \frac{k_1}{k+5},$$

$$k_a = (a_1 - a)(a_2 - a)(a_3 - a).$$

A rekurziós képlet megoldása csökkenő k=N,N-1,...1,0 mentén történik, normalizációval; c_k-hoz SG=1, d_k-hoz SG=-1.

A FORTRAN lista:

```
      SUBROUTINE SHIP32(A1,A2,A3,B1,B2,RM,K,N,E,SG)
      INTEGER SG
      DOUBLE PRECISION A1,A2,A3,B1,B2,RM,E,S,R
      DIMENSION E(N),S(23)
      COMMON/MUNK/R(612)
      DO 1 I=1,5
1     S(I)=I
      S(6)=(S(1)-S(2)/B1)*(S(1)-S(2)/B2)
      S(7)=(S(1)-S(1)/A1)*(S(1)-S(1)/A2)*(S(1)-S(1)/A3)
      S(8)=(S(1)-S(2)/A1)*(S(1)-S(2)/A2)*(S(1)-S(2)/A3)
      S(9)=A1+A2+A3
      S(1Ø)=2*B1/A1*B2/A2/RM/A3
      S(11)=S(1)/B1/B2
      S(12)=((S(1)/A1)/A2)/A3
      S(23)=K-1
      R(K+1)=1.ØD-7Ø
      DO 5 I=2,6
5     R(K+I)=Ø.ØD-Ø
      DO 6 I=1,K
      S(16)=S(6)/(S(23)+S(2))
      S(17)=S(6)/(S(23)+S(4))
      S(13)=S(16)+(S(23)+B1+B2-2)*S(11)
      S(14)=-S(16)-S(17)+2*(S(23)+3)*S(11)
      S(15)=S(17)+(S(23)-B1-B2+8)*S(11)
      S(17)=S(7)/(S(23)+S(1))
      S(18)=S(8)/(S(23)+S(2))
      S(19)=S(8)/(S(23)+S(4))
      S(2Ø)=S(7)/(S(23)+S(5))
      S(16)=S(17)-S(18)+(S(23)+S(9)-3)*S(12)
      S(17)=S(19)-S(17)+(3*S(23)+S(9)+3)*S(12)
      S(18)=S(18)-S(2Ø)+(S(23)*3-S(9)+15)*S(12)
```

```
      S(19)=S(20)-S(19)+(S(23)-S(9)+9)*S(12)
      J=K+1-I
      R(J)=S(10)*(S(13)*R(J+1)+SG*S(14)*R(J+3)+S(15)*R(J+5))
      R(J)=(R(J)-SG*S(17)*R(J+2)-S(18)*R(J+4)-SG*S(19)*R(J+6))/S(16)
6     S(23)=S(23)-S(1)
      R(1)=R(1)/S(2)
      S(13)=R(1)
      S(14)=-SG
      DO 7 I=3,K,2
      S(13)=S(13)+S(14)*R(I)
7     S(14)=-S(14)*SG
      DO 8 I=1,K
8     R(I)=R(I)/S(13)
21    DO 10 I=1,N
10    E(I)=R(I)
13    RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SHIP32(A1,A2,A3,B1,B2,RM,K,N,E,SG)

A1,A2,A3 számláló paraméterek $\neq$ 0,-1,-2,... /bemenő/,

B1,B2 nevező paraméterek $\neq$ 0,-1,-2,... /bemenő/,

RM a függvény argumentuma= λ /bemenő/,

K az iteráció kezdő értéke /bemenő/,

N az együtthatók száma /bemenő/,

E N méretü tömb, tartalma $c_0, c_1, \ldots, c_{N-1}$ vagy $d_0, d_1, \ldots, d_{N-1}$ /kimenő/,

SG értéke = 1 c_k együtthatókra, értéke = -1 d_k együtthatókra /bemenő/.

Megjegyzések:

A nyert sorozat pontosságáról ugy nyerhetünk információt, hogy elvégeztetjük a számitást egy konkrét K értékkel, majd pl.: a K'=K+10 értékkel és a két eredményt összehasonlitjuk. A K értéke 612 alatt kell legyen.

A szubrutin az ${}_3F_2$ függvényen kivül alkalmas az ${}_2F_2$, ${}_2F_1$, ${}_1F_2$, ${}_1F_1$, ${}_0F_2$, ${}_0F_1$ függvények megfelelő együtthatóinak generálására a konfluencia elv alapján. Ez abban áll, hogy az ${}_3F_2$ függvény paramétereinek és argumentumának megfelelő végtelenhez tartása esetén egy másik hipergeometriai függvényhez konvergál, pl.:

$$\lim_{a_1 \to \infty} {}_3F_2\left(a_1, a_2, a_3; b_1, b_2; \frac{\lambda x}{a_1}\right) = {}_2F_2(a_2, a_3; b_1, b_2; \lambda x)$$

A szubrutinban az $a_1 = 10^{25}$ helyettesités megvalósitja az $a_1 \to \infty$ határátmenetet.

A K értékét ajánlatos a függvény analitikai sajátságainak megfelelően választani, durván szólva, elégséges a K=N+10 választás.

d4/ A SHIP43 program.

LEIRÁS

Az

$$f_1(x) = {}_4F_3(a_1, a_2, a_3, a_4; b_1, b_2, b_3; \lambda x),$$

Csebisev sorának

$$f_1(x) = \sum_{k=0}^{\infty} C_k T_k(x),$$

együtthatóit, valamint az

$$f_2(x) = \mathrm{Re}\, f_1(ix), \quad f_3(x) = \mathrm{Im}\, f_1(ix),$$

függvények

$$f_2(x) = \sum_{k=0}^{\infty} d_{2k} (-1)^k T_{2k}(x),$$

$$f_3(x) = 2\sum_{k=0}^{\infty} d_{2k+1} (-1)^k T_{2k+1}(x)$$

Csebisev sorainak együtthatóit határozza meg.

A számitás rekurziós képletet használ:

$$\frac{2}{\lambda}\left(B_0 c_{k+1} + SG\cdot B_1 c_{k+3} + B_2 c_{k+5} + SG\cdot B_3 c_{k+7}\right) =$$

$$= A_0 c_k + SG\cdot A_1 c_{k+2} + A_2 c_{k+4} + SG\cdot A_3 c_{k+6} + A_4 c_{k+8},$$

$$k = 0, 1, 2, \ldots$$

$$B_0 = k + p - 5 + \frac{k_2}{k+2} - \frac{k_3}{k+3},$$

$$B_1 = 3k + p + 3 - \frac{k_2}{k+2} + \frac{k_3}{k+5},$$

$$B_2 = 3k - p + 21 + \frac{k_3}{k+3} - \frac{k_2}{k+6},$$

$$B_3 = k - p + 13 - \frac{k_3}{k+5} + \frac{k_2}{k+6},$$

$$p = b_1 + b_2 + b_3, \quad k_a = (b_1 - a)(b_2 - a)(b_3 - a),$$

$$A_0 = k + q - 6 - \frac{l_1}{k+1} - \frac{2l_2}{k+2} + \frac{l_3}{k+3},$$

$$A_1 = 4k + 2q - 4 - \frac{l_1}{k+1} + \frac{l_1 + l_3}{k+3} - \frac{l_3}{k+5},$$

$$A_2 = 6k + 24 + \frac{2l_2}{k+2} - \frac{l_1 + l_3}{k+3} - \frac{l_1 + l_3}{k+5} + \frac{2l_2}{k+6},$$

$$A_3 = 4k - 2q + 36 - \frac{l_3}{k+3} + \frac{l_1 + l_3}{k+5} - \frac{l_1}{k+7},$$

$$A_4 = k - q + 14 + \frac{l_3}{k+5} - \frac{2l_2}{k+6} + \frac{l_1}{k+7},$$

$$q = a_1 + a_2 + a_3 + a_4,$$

$$l_a = \frac{1}{2}(a_1 - a)(a_2 - a)(a_3 - a)(a_4 - a).$$

A rekurziós képlet megoldása csökkenő k=N,N-1,...,1,0 mentén történik, normalizációval; c_k-hoz SG=1, d_k-hoz SG=-1.

A FORTRAN lista:

```
      SUBROUTINE SHIP43(A1,A2,A3,A4,B1,B2,B3,RM,K,N,E,SG)
      INTEGER SG
      DOUBLE PRECISION A1,A2,A3,A4,B1,B2,B3,RM,E,S,R
      DIMENSION E(N),S(28)
      COMMON/MUNK/R(612)
      S(2)=(1-2/B1)*(1-2/B2)*(1-2/B3)
      S(3)=(1-3/B1)*(1-3/B2)*(1-3/B3)
      S(4)=(1-1/A1)*(1-1/A2)*(1-1/A3)*(1-1/A4)/2
      S(5)=(1-2/A1)*(1-2/A2)*(1-2/A3)*(1-2/A4)
      S(6)=(1-3/A1)*(1-3/A2)*(1-3/A3)*(1-3/A4)/2
      S(7)=B1+B2+B3
      S(8)=A1+A2+A3+A4
      S(9)=2/RM
      S(10)=((1/B1)/B2)/B3
      S(11)=(((1/A1)/A2)/A3)/A4
      S(1)=K-1
      R(K+1)=1.0D-70
      DO 2 I=2,8
2     R(K+I)=0.0D-0
      DO 5 I=1,K
      S(16)=S(2)/(S(1)+2)
      S(17)=S(2)/(S(1)+6)
      S(18)=S(3)/(S(1)+3)
      S(19)=S(3)/(S(1)+5)
      S(12)=S(16)-S(18)+(S(1)+S(7)-5)*S(10)
      S(13)=S(19)-S(16)+(3*S(1)+S(7)+3)*S(10)
      S(14)=S(18)-S(17)+(3*S(1)-S(7)+21)*S(10)
      S(15)=S(17)-S(19)+(S(1)-S(7)+13)*S(10)
      S(20)=S(4)/(S(1)+1)
      S(21)=S(4)/(S(1)+7)
      S(22)=S(5)/(S(1)+2)
      S(23)=S(5)/(S(1)+6)
      S(24)=S(6)/(S(1)+3)
      S(25)=S(6)/(S(1)+5)
      S(26)=S(4)/(S(1)+3)
      S(27)=S(4)/(S(1)+5)
      S(16)=S(24)-S(20)-S(22)+(S(1)+S(8)-6)*S(11)
      S(17)=-S(20)+S(26)+S(24)-S(25)+(4*S(1)+2*S(8)-4)*S(11)
      S(18)=S(22)-S(26)-S(24)-S(27)-S(25)+S(23)+(S(1)+4)*6*S(11)
```

```
      S(19)=S(27)+S(25)-S(24)-S(21)+(4*S(1)-2*S(8)+36)*S(11)
      S(22)=S(25)-S(23)+S(21)+(S(1)-S(8)+14)*S(11)
      J=K+1-I
      R(J)=S(9)*(S(12)*R(J+1)+SG*S(13)*R(J+3)+S(14)*R(J+5))
      R(J)=R(J)+S(9)*SG*S(15)*R(J+7)-SG*S(17)*R(J+2)-S(18)*R(J+4)
      R(J)=(R(J)-SG*S(19)*R(J+6)-S(20)*R(J+8))/S(16)
5     S(1)=S(1)-1
      R(1)=R(1)/2
      S(24)=R(1)
      S(25)=-SG
      DO 6 I=3,K,2
      S(24)=S(24)+S(25)*R(I)
6     S(25)=-SG*S(25)
      DO 7 I=1,K
7     R(I)=R(I)/S(24)
21    DO 9 I=1,N
9     E(I)=R(I)
13    RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SHIP43(A1,A2,A3,A4,B1,B2,B3,RM,K,N,E,SG)

A1,A2,A3,A4 számláló paraméterek $\neq$ 0,-1,-2,... /bemenő/,

B1,B2,B3 nevező paraméterek $\neq$ 0,-1,-2,... /bemenő/,

RM az argumentum $= \lambda \cdot \dfrac{A1.A2.A3.A4}{B1.B2.B3}$ /bemenő/,

K az iteráció kezdő értéke /bemenő/,

N az együtthatók száma /bemenő/,

E N méretü tömb, tartalma, $c_0, c_1, \dots, c_{N-1}$ vagy $d_0, d_1, \dots, d_{N-1}$ /kimenő/,

SG értéke = 1 c_k együtthatókra, értéke = -1 d_k együtthatókra /bemenő/.

Megjegyzések:

A nyert sorozat pontosságáról ugy nyerhetünk információt, hogy elvégeztetjük a számitást egy konkrét K értékkel, majd pl.: a K'=K+10 értékkel és a két eredményt összehasonlitjuk. A K értéke 612 alatt kell legyen.

A szubrutin az ${}_4F_3$ függvényen kivül alkalmas az ${}_3F_3$, ${}_3F_2$, ${}_2F_2$, ${}_2F_1$, ${}_1F_3$, ${}_1F_2$, ${}_1F_1$, ${}_0F_3$, ${}_2F_3$, ${}_0F_2$, ${}_0F_1$ függvények megfelelő együtthatóinak generálására a konfluencia elv alapján. Ez abban áll, hogy az ${}_4F_3$ függvény paramétereinek és argumentumának megfelelő végtelenhez tartása esetén egy másik hipergeometriai függvényhez konvergál, pl.:

$$\lim_{a_4 \to \infty} {}_4F_3(a_1, a_2, a_3, a_4; b_1, b_2, b_3; \lambda x/a_4) = {}_3F_3(a_1, a_2, a_3; b_1, b_2, b_3; \lambda x)$$

A szubrutinban az $a_4 = 10^{25}$ helyettesités megvalósitja az $a_4 \to \infty$ határátmenetet.

A K értéket ajánlatos a függvény analitikai sajátságainak megfelelően választani, durván szólva, elégséges a K=N+10 választás.

e/ Polinomokkal való müveletek és egyébb segédprogramok.

Közelitések készitésekor szükség van polinomokkal való müveletek elvégzésére. Ehhez nyujtanak segitséget az EXPPOL, LOGPOL,HATPOL,POLPPL,BEHPOL,SORINV szubrutinok.

A PRAKA csomag szubrutinjainak végrehajtásához nyujtanak segitséget a RENDZ1,RENDZ2,RENDZ3,MAXKER,MEGOLD,CSESUM,EGYUTT, SZEGYU szubrutinok.

e1/ Az EXPPOL program.

LEIRÁS

Az

$$E(x) = e^{P(x)} = \sum_{k=0}^{\infty} c_k x^k$$

hatványsorának első N tagját számitja ki, ahol $P(x)$ adott polinom $P(0)=0$

$$P(x) = \sum_{i=1}^{N-1} a_i x^i.$$

A számitás az alábbi képletet használja:

$$(k+1)c_k = \sum_{j=0}^{k} (j+1)a_{j+1}c_{k-j}\ , \quad k=1,2,\dots,$$

$$c_0 = 1\ , \quad c_1 = a_1 .$$

A FORTRAN lista:

```
      SUBROUTINE EXPPOL(A,N,B)
      DOUBLE PRECISION A,B,C
      DIMENSION A(N),B(N)
      B(1)=1.0D-0
      N1=N-1
      DO 1 I=1,N1
      C=0.0D-0
      DO 2 K=1,I
2     C=C+B(K)*(I+B(1)-K)*A(I-K+2)
1     B(I+1)=C/I
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL EXPPOL(A,N,B)

A,B N méretü tömbök,

A tartalma: $a_0 = 0,\ a_1, \dots,\ a_{N-1}$ a P polinom együtthatói /bemenő/,

N az együtthatók száma /bemenő/,

B tartalma: $c_0, c_1, \dots, c_{N-1}$ az E polinom együtthatói /kimenő/.

e2/ A LOGPOL program.

LEIRÁS

Az

$$\ln P(x) = \sum_{i=1}^{\infty} c_i x^i$$

hatványsorát számolja. $P(x)$ adott $(P(0)=1)$ polinom:

$$P(x) = 1 + \sum_{i=1}^{N-1} a_i x^i .$$

A számitás az alábbi képletet használja:

$$c_{k+1} = a_{k+1} + \frac{1}{k+1} \sum_{j=1}^{k} (k-j+1) c_{k-j+1} a_j$$

$$k = 1, 2, \dots, \quad c_1 = a_1 .$$

A FORTRAN lista:

```
      SUBROUTINE LOGPOL(A,N,B)
      DOUBLE PRECISION A,B,C
      DIMENSION A(N),B(N)
      B(1)=Ø,ØD-Ø
      B(2)=A(2)
      N1=N-2
      DO 1 I=1,N1
      C=Ø,ØD-Ø
      DO 2 K=1,I
2     C=C-K*B(K+1)*A(I-K+2)
1     B(I+2)=A(I+2)+C/(I*A(1))
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL LOGPOL(A,N,B)

A,B N méretü tömbök

A tartalma: $a_0 = 1, a_1, \dots, a_{N-1}$ a P polinom együtthatói; /bemenő/,

N az együtthatók száma /bemenő/,

B tartalma: $c_0 = 0, c_1, \dots, c_{N-1}$ ln P polinom együtthatói /bemenő/.

e3/ A HATPOL program.

LEIRÁS

A

$$H(x) = \{P(x)\}^{\alpha} = \sum_{i=0}^{\infty} c_i x^i ,$$

hatványsorát számolja. $P(x)$ adott $N-1$ foku polinom:

$$P(x) = \sum_{i=0}^{N-1} a_i x^i , \qquad \alpha \neq 0 \text{ konstans.}$$

A számitás az alábbi képletet használja.

$$c_{k+1} = \alpha \frac{c_0}{a_0} a_{k+1} + \frac{1}{a_0(k+1)} \sum_{j=1}^{k} a_{k+1-j} c_j [\alpha(k+1-j)-j],$$

$$c_0 = a_0^{\alpha}, \quad c_1 = \alpha\, a_0^{\alpha-1} \cdot a_1 \quad , k = 1, 2, \ldots$$

A FORTRAN lista:

```
      SUBROUTINE HATPOL(A,N,ALFA,B)
      DOUBLE PRECISION A,B,ALFA,C,D,E,F,G
      DIMENSION A(N),B(N)
      E=ALFA
      C=1.0D-0
      B(1)=A(1)**E
      D=E*B(1)/A(1)
      B(2)=D*A(2)
      N1=N-2
      DO 1 I=1,N1
      F=D*A(I+2)
      G=0.0D-0
      DO 2 K=1,I
2     G=G+B(K+1)*A(I-K+2)*(E*(I+C-K)-K)
1     B(I+2)=F+G/A(1)/(I+C)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL HATPOL(A,N,ALFA,B)

A,B N méretü tömbök,

A tartalma: $a_0, a_1, \ldots, a_{N-1}$ a P polinom együtthatói /bemenő/,

N az együtthatók száma /bemenő/,

ALFA= $\alpha \neq 0$, /bemenő/,

B tartalma: $c_0, c_1, \ldots, c_{N-1}$ a H polinom együtthatói /kimenő/.

e4/ A POLPPL program.

LEIRÁS

Az

$$f(x)=\frac{\sum_{i=0}^{N-1} a_i x^i}{\sum_{i=0}^{N-1} b_i x^i}=\sum_{i=0}^{\infty} c_i x^i$$

törtfüggvény hatványsorát számitja ki.

A számitás az alábbi képletet használja:

$$c_0=\frac{a_0}{b_0}$$

$$c_k=\frac{1}{b_0}\left(a_k-\sum_{j=1}^{k} c_{k-j} b_j\right),\quad k=1,2,\ldots$$

A FORTRAN lista:

```
      SUBROUTINE POLPPL(A,B,N,C)
      DOUBLE PRECISION A,B,C,D
      DIMENSION A(N),B(N),C(N)
      C(1)=A(1)/B(1)
      N1=N-1
      DO 1 I=1,N1
      D=A(I+1)
      DO 2 K=1,I
    2 D=D-C(K)*B(I+2-K)
    1 C(I+1)=D/B(1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL POLPPL(A,B,N,C)

A,B,C N méretü tömbök,

A tartalma: $a_0, a_1, \ldots, a_{N-1}$ számok /bemenő/,

B tartalma: $b_0, b_1, \ldots, b_{N-1}$ számok /bemenő/,

N az együtthatók száma /bemenő/,

C tartalma: $c_0, c_1, \ldots, c_{N-1}$ számok /kimenő/.

Megjegyzés:

A program nem vizsgálja a megadott a_i ill. b_i számokat, pl.: ha $b_0=0$, C tartalma hibás lesz.

e5/ A BEHPOL program.

LEIRÁS

Az

$$f(x)=P_1(P_2(x))=\sum_{i=0}^{\infty} c_i x^i$$

hatványsort számitja ki, $P_1(x)$ és $P_2(x)$ adott

$$P_1(x)=\sum_{i=0}^{N-1} a_i x^i,$$

$$P_2(x)=\sum_{i=0}^{N-1} b_i x^i$$

polinomok.

A számitás módszere a közvetlen behelyettesités.

Csupán az első N tagot számitja ki.

A FORTRAN lista:

```
      SUBROUTINE BEHPOL(A,B,N,C)
      DOUBLE PRECISION A,B,C,D,R
      DIMENSION A(N),B(N),C(N)
      COMMON/MUNK/R(612)
      DO 1 I=1,N
      C(I)=A(2)*B(I)
 1    R(I)=B(I)
      N1=N-2
      DO 2 J=1,N1
      DO 3 I=1,N
 3    R(N+I)=0.0D-0
      DO 4 I=1,N
      NI=N-I+1
      DO 4 K=1,NI
 4    R(N+I+K-1)=R(N+I+K-1)+R(K)*B(I)
      DO 2 I=1,N
      R(I)=R(N+I)
 2    C(I)=C(I)+R(I)*A(J+2)
      C(1)=C(1)+A(1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL BEHPOL(A,B,N,C)

A,B,C N méretü tömbök,

A tartalma: $a_0, a_1, \ldots, a_{N-1}$ a P_1 polinom együtthatói /bemenő/,

B tartalma: $b_0, b_1, \ldots, b_{N-1}$ a P_2 polinom együtthatói /bemenő/,

N az együtthatók száma /bemenő/,

C tartalma: $c_0, c_1, \ldots, c_{N-1}$ $f(x)$ együtthatói /kimenő/.

e6/ A SORINV program.

LEIRÁS

A

$$t = x\,P(t)$$

egyenlet

$$t = \sum_{k=1}^{\infty} c_k x^k$$

alaku megoldásának sorát számitja ki. A $P(x)$ polinom:

$$P(x) = \sum_{i=0}^{N-1} a_i x^i$$

A számitás az alábbi képletet használja:

$$c_k = \frac{1}{k!}\,\frac{d^{k-1}}{dt^{k-1}}\left\{\left[P(t)\right]^k\right\}\Big|_{t=0}.$$

A megoldás módszere a közvetlen behelyettesités.

A FORTRAN lista:

```
      SUBROUTINE SORINV(A,N,B)
      DOUBLE PRECISION A,B,R
      DIMENSION A(N),B(N)
      COMMON/MUNK/R(612)
      DO 4 I=1,N
 4    R(I)=A(I)
      N1=N-1
      DO 1 J=1,N1
      DO 3 I=1,N
 3    R(N+I)=0.0D-0
      DO 2 I=1,N
      NI=N-I+1
      DO 2 K=1,NI
 2    R(N+K+I-1)=R(N+K+I-1)+R(K)*A(I)
      DO 5 I=1,N
 5    R(I)=R(N+I)
 1    R(J+1)=R(J+1)/(J+1)
      R(1)=A(1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SORINV(A,N,B)

A,B N méretü tömbök,

A tartalma: $a_0, a_1, \ldots, a_{N-1}$ a P polinom együtthatói /bemenő/,

N az együtthatók száma /bemenő/,

B tartalma: $c_1, c_2, \ldots c_{N-1}, c_N$ /kimenő/.

e7/ A RENDZ1 program.

LEIRÁS

Az

$$f(x) = b_0 + \cfrac{a_1 x}{b_1 + \cfrac{a_2 x}{b_2 + \ldots \cfrac{a_{n-1} x}{b_{n-1}}}},$$

lánctörtet algebrai polinomok hányadosává alakitja át.

$$f(x) = \frac{\sum_{i=0}^{K} u_i x^i}{1 + \sum_{i=1}^{M} v_i x^i} .$$

A megoldás módszere a rekurziv kiszámitás.

A FORTRAN lista:

```
      SUBROUTINE RENDZ1(A,B,N,E1,E2,R)
      DOUBLE PRECISION A,B,E1,E2,R,W,T
      DIMENSION A(N),B(N),E2(N),E1(N),R(N)
      DO 1 I=1,N
      E1(I)=0,0D-0
      E2(I)=0,0D-0
1     R(I)=0,0D-0
      E2(1)=B(N)
      R(1)=1 0D-0
      N1=N-1
      DO 2 I=1,N1
      W=B(N-I)
      T=A(N+1-I)
      E1(1)=W*E2(1)
      IB=(I+3)/2
      DO 3 K=2,IB
3     E1(K)=W*E2(K)+T*R(K-1)
      DO 2 J=1,IB
      R(J)=E2(J)
2     E2(J)=E1(J)
      T=R(1)
      NF=1+(N+1)/2
      DO 4 I=1,NF
      E1(I)=E1(I)/T
4     E2(I)=R(I)/T
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL RENDZ1(A,B,N,E1,E2,R)

A,B,E1,E2,R N méretü tömbök,

A tartalma: $a_0=0, a_1, \dots, a_{N-1}$ /bemenő/,

B tartalma: $b_0, b_1, \dots, b_{N-1}$ /bemenő/,

N az együtthatók száma /bemenő/,

E1 tartalma: $u_0, u_1, \dots,$, számláló együtthatói /kimenő/,

E2 tartalma: $v_0, v_1, \dots,$, nevező együtthatói /kimenő/,

R munkatömb, a tartalma közönbös /be-, kimenő/.

e8/ A RENDZ2 program.

LEIRÁS

Az

$$f(x) = c_0 + c_1(x-x_0) + \dots + c_{\ell-m}(x-x_0)(x-x_1)\dots(x-x_{\ell-m-1}) +$$

$$+ \cfrac{(x-x_0)(x-x_1)\cdots(x-x_{\ell-m})}{c_{\ell-m+1} + \cfrac{x-x_{\ell-m+1}}{c_{\ell-m+2} + \dots + \cfrac{x-x_{\ell+m-1}}{c_{\ell+m}}}}.$$

kifejezést átirja algebrai polinomok hányadosává:

$$f(x) = \frac{\sum_{i=0}^{L} a_i x^i}{1 + \sum_{i=1}^{M} b_i x^i}.$$

A FORTRAN lista:

```
      SUBROUTINE RENDZ2(C,X,E,W,N,M)
      DOUBLE PRECISION C,X,E,W,A,B,D
      DIMENSION C(N),X(N),E(N),W(3)
      COMMON/WORK2/A(50),B(50),D(50)
      M1=M+1
      DO 1 I=1,M1
      A(I)=0.0D-0
      B(I)=0.0D-0
1     D(I)=0.0D-0
      B(1)=C(N)
      D(1)=1.0D-0
      MM=2*M-1
      DO 2 I=1,MM
      W(1)=C(N-I)
      W(2)=X(N-I)
      A(1)=W(1)*B(1)-W(2)*D(1)
      II=(I+3)/2
      DO 3 K=2,II
3     A(K)=W(1)*B(K)+D(K-1)-W(2)*D(K)
      DO 2 J=1,II
      D(J)=B(J)
2     B(J)=A(J)
      K=N-2*M
      DO 4 I=1,K
4     E(I)=C(I)
      IF(K-1) 11,11,12
12    KK=K-1
```

```
      DO 5 J=1,KK
      KKK=K-J
      W(1)=X(KKK)
      DO 5 L=KKK,KK
5     E(L)=E(L)-W(1)*E(L+1)
11    NM=N-M
      DO 13 I=1,NM
13    A(I)=0.0D-0
      DO 6 I=1,M1
      W(1)=B(I)
      DO 6 L=1,K
      I1=L+I-1
6     A(I1)=A(I1)+E(L)*W(1)
      DO 16 I=1,K
      MI=M+I-1
      D(MI+1)=D(MI)
      IF(MI-1) 14,14,15
15    DO 7 L=2,MI
      LB=MI+2-L
7     D(LB)=D(LB-1)-X(I)*D(LB)
14    D(1)=-X(I)*D(1)
16    CONTINUE
      DO 8 I=1,NM
8     E(I)=A(I)+D(I)
      DO 9 I=1,M
9     E(N-M+I)=B(I+1)
      DO 10 I=1,N
10    E(I)=E(I)/B(1)
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL RENDZ2(C,X,E,N,M)

C,X,E N méretü tömbök,

C tartalma: $c_0, c_1, \dots, c_{N-1}$, N=L+M+1, $L \geq M$ /bemenő/,

X tartalma: $x_0, x_1, \dots, x_{N-1}$ alappontok /bemenő/,

E tartalma: a számláló és nevező együtthatók, folytonosan elhelyezve, $a_0, a_1, \dots, a_{L-1}, b_1, b_2, \dots, b_M$ /kimenő/,

N az együtthatók száma /bemenő/,

M a nevező polinom foka /bemenő/.

A program használ egy WORK2 nevü COMMON mezőt, amelynek tartalma három A,B,D 50 méretü tömb /munkatömbök/.

e9/ A RENDZ3 program.

LEIRÁS

Az

$$R_{N,M}(x)=\frac{\sum_{i=0}^{N-1} a_i p_i(x)}{\sum_{i=0}^{M-1} b_i p_i(x)}$$

kifejezést, ahol $p_i(x)$ vagy $T_i(x)$ vagy $T_i^*(x)$ átrendezi

$$R_{N,M}(x)=\frac{\sum_{i=0}^{N-1} u_i x^i}{\sum_{i=0}^{M-1} v_i x^i}$$

algebrai polinomok hányadosává.

A FORTRAN lista:

```
      SUBROUTINE RENDZ3(A,N,B,M,L,W)
      DOUBLE PRECISION A,B,W,S
      DIMENSION A(N),B(M),W(N)
      IF(L) 1,1,2
1     CALL SZCSTA(B,W,M)
      S=W(1)
      DO4 I=1,M
4     B(I)=W(I)/S
      CALL SZCSTA(A,W,N)
      DO 3 I=1,N
3     A(I)=W(I)/S
      GO TO 7
2     CALL CSETAY(B,W,M)
      S=W(1)
      DO 6 I=1,M
6     B(I)=W(I)/S
      CALL CSETAY(A,W,N)
      DO 5 I=1,N
5     A(I)=W(I)/S
7     RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL RENDZ3(A,N,B,M,L,W)

A,B,W N méretü tömbök,

A tartalma: $a_0, a_1, \ldots, a_{N-1}$, majd $u_0, u_1, \ldots, u_{N-1}$ /bemenő és kimenő/,

N a számláló együtthatók száma /bemenő/,

B tartalma: $b_0, b_1, \ldots, b_{M-1}$ majd $v_0 = 1, v_1, \ldots, v_{M-1}$ /bemenő és kimenő/,

M a nevező együtthatók száma /bemenő/,

L, $T_k(x)$ -re L=-1, $T_k^*(x)$ -re L=1 /bemenő/,

W munkatömb, tartalma közömbös /be-, kimenő/.

e10/ A MAXKER program.

LEIRÁS

Adva van három pont x_1, x_2, x_3 a nekik megfelelő függvény értékekkel és közelitő függvény értékekkel: f_1, s_1; f_2, s_2; f_3, s_3. A program a három érték közül kiválasztja azt, amelyre $f_i - s_i$ abszolut értékben maximális és előjele megegyezik $f_2 - s_2$ előjelével. Az eredmény X és F-be kerül.

A FORTRAN lista:

```
      SUBROUTINE MAXKER(X1,F1,S1,X2,F2,S2,X3,F3,S3,X,F)
      DOUBLE PRECISION X1,F1,S1,X2,F2,S2,X3,F3,S3,X,F,T,T1
      X=X2
      F=F2
      T1=S2
      T=(F-T1)/(F1-S1)
      IF(T) 4,3,1
1     IF(T-1.ØD-Ø) 3,4,4
3     X=X1
      F=F1
      T1=S1
4     T=(F-T1)/(F3-S3)
      IF(T) 5,6,2
2     IF(T-1.ØD-Ø) 6,5,5
6     X=X3
      F=F3
5     RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL MAXKER(X1,F1,S1,X2,F2,S2,X3,F3,S3,X,F)

A paraméter lista elemei mind dupla preciziós változók,

X1,F1,S1 alappont, függvényérték, közelitő függvényérték, /bemenő/,

X2,F2,S2 -"- , -"- , -"- , /bemenő/,

X3,F3,S3 -"- , -"- , -"- , /bemenő/,

X,F a maximális tulajdonságu alappont és függvényérték /kimenő/.

Ez a szubrutin a MINMAX és MINRAC szubrutinok számára hasznos, amikor a kvadratikus interpoláció durva.

e11/ A MEGOLD program.

LEIRÁS

A program lineáris egyenletrendszert old meg főelem kiválasztásos Gauss elimimációval.

A kétdimenziós A tömbben helyezendő el az egyenletrendszer mátrixa, az utolsó oszlopban a szabad tagok. A megoldás ugyan ennek a mátrixnak utolsó sorában helyezkedik el.

A FORTRAN lista:

```
      SUBROUTINE MEGOLD(A,B,N)
      DOUBLE PRECISION A,B,T,S
      DIMENSION A(N,N),B(N,N)
      L=N-1
      DO 1 I=1,L
 1    B(I,1)=A(I,N)
      DO 2 K=1,L
      DO 2 I=1,L
 2    B(I,K+1)=A(I,K)
      DO 3 K=2,L
      M=N+2-K
      T=B(K-1,M)
      IF(T) 6,6,7
 6    T=-T
 7    JB=K-1
      DO 16 JJ=K,L
      S=B(JJ,M)
      IF(S) 11,11,12
 11   S=-S
```

```
12      IF(T-S) 13,13,16
13      T=S
        JB=JJ
1Ø      CONTINUE
16      CONTINUE
        IF(T) 14,33,14
14      DO 15 II=1,M
        S=B(JB,II)
        B(JB,II)=B(K-1,II)
15      B(K-1,II)=S
        S=B(K-1,M)
        DO 3 J=K,L
        T=B(J,M)/S
        M1=M-1
        DO 3 I=1,M1
3       B(J,M-I)=B(J,M-I)-T*B(K-1,M-I)
        A(N,1)=B(L,1)/B(L,2)
        DO 4 I=2,L
        T=Ø,ØD-Ø
        I1=I-1
        DO 5 J=1,I1
5       T=T+A(N,J)*B(N-I,J+1)
4       A(N,I)=(B(N-I,1)-T)/B(N-I,I+1)
33      RETURN
        END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL MEGOLD(A,B,N)

A,B kétdimenziós N*N méretü tömbök,

A tartalma: az együttható mátrix, szabad tagokkal, a megoldás e mátrix utolsó sorában lesz /bemenő, kimenő/,

B munkatömb /be-, kimenő/,

N = n+1, n az egyenletek száma /bemenő/.

Megjegyzések:

A szubrutin nem vizsgálja meg, hogy a rendszer megoldható-e vagy sem. Ellenben, ha felfedezi, hogy a rendszer determinánsa zérus, a további munkát befejezi és visszaadja a vezérlést a hivó szubrutinnak /természetesen hibás végeredménnyel/.

A MEGOLD szubrutint a PADEMK és CSPAMK szubrutinok használják, de a felhasználó főprogramja önállóan is használhatja. Fel van téve, hogy $n \geqslant 2$.

e12/ A CSESUM program.

LEIRÁS

E szubrutin Csebisev polinomok szerint rendezett összeg adott argumentum értéknél való kiszámitását végzi:

$$f(x) = \sum_{i=0}^{N-1} a_i T_i^*(x).$$

A kiszámitás módszere az alábbi rekurziv séma:

$$f(x) = b_1 - (2x-1)\, b_2,$$

$$b_{N+1} = 0, \qquad b_{N+2} = 0,$$

$$b_n = 2(2x-1)\, b_{n+1} - b_{n+2} + a_n,$$

$$n = N, N-1, \ldots, 2, 1.$$

A FORTRAN lista:

```
      SUBROUTINE CSESUM(X,E,N,S)
      DOUBLE PRECISION X,E,A,B,S,P
      DIMENSION E(N)
      P=4*X-2
      A=E(N)
      B=0.0D-0
      I1=N-1
      DO 1 I=1,I1
      S=P*A-B+E(N-I)
      B=A
    1 A=S
      S=A-P*B/2
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL CSESUM(X,E,N,S)

X argumentum /bemenő/,

E N méretü tömb, tartalma: $a_0, a_1, \ldots, a_{N-1}$ /bemenő/,

N az együtthatók száma /bemenő/,

S $f(x)$ értéke /kimenő/.

e13/ Az EGYUTT program.

LEIRÁS

Kiszámitja a $T_i^*(x)$ Csebisev polinom együtthatóit $i=0,1,\ldots,$ N-re és folytonosan elhelyezi a COEFF nevü COMMON mező A tömbjébe:

$$T_i^*(x) = \sum_{j=0}^{i} C_j^{(i)} x^j \quad , \ 0 \le x \le 1,$$

ahol

$$C_j^{(i)} = (-1)^{i+j} 4^j \frac{i\Gamma(i+j)}{\Gamma(2j+1)\Gamma(i-j+1)} .$$

A FORTRAN lista:

```
      SUBROUTINE EGYUTT(N)
      DOUBLE PRECISION A,S,P
      COMMON /COEFF/A(496)
      A(1)=1.0D-0
      A(2)=-1.0D-0
      A(3)=2.0D-0
      K=4
      S=1.0D-0
      DO 1 I=2,N
      P=I*I
      A(K)=S
      DO 2 J=1,I
      L=K+J
      L1=L-1
2     A(L)=-A(3)*(P-(J-1)**2)/J/(A(3)*J-1)*A(L1)
      S=-S
1     K=L+1
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL EGYUTT(N)

N paraméter adja meg a maximális fokszámot, ameddig ki kell számitani /és elhelyezni a COEFF nevü COMMON mező A tömbjébe/ a Csebisev polinomok együtthatóit. $N \leq 30$ /bemenő/.

e14/ A SZEGYU program.

LEIRÁS

Kiszámitja a $T_i(x)$ Csebisev polinom együtthatóit $i = 0,1,2,\ldots$ N-re és elhelyezi folytonosan a COEFF nevü COMMON mező A tömbjébe:

$$T_i(x) = \sum_{j=0}^{[\frac{i}{2}]} d_j^{(i)} x^{i-2j}, \quad -1 \leq x \leq 1,$$

ahol

$$d_j^{(i)} = (-1)^j 4^j \frac{i\,\Gamma(i-j)}{\Gamma(i-2j+1)\,\Gamma(j+1)}$$

A FORTRAN lista:

```
      SUBROUTINE SZEGYU(N)
      DOUBLE PRECISION A,S,P
      COMMON/COEFF/A(496)
      A(1)=1.0D-0
      NK=(N*(N+1))/2
      DO 1 I=2,NK
1     A(I)=0.0D-0
      A(3)=A(1)
      S=2.0D-0
      P=S
      N1=N-1
      DO 2 I=2,N1
      K=((I+1)*(I+2))/2
      A(K)=P
      DO 3 J=2,I,2
      KJ=K-J
3     A(KJ)=-A(KJ+2)*(I-J+1)*(I-J+2)/(2*I-J)/J
2     P=P*S
      RETURN
      END
```

A SZUBRUTIN HIVÁSA, PARAMÉTEREK.

CALL SZEGYU(N)

N paraméter adja meg azt a maximális fokszámot, ameddig ki kell számitani /és elhelyezni a COEFF nevü COMMON mező A tömbjébe/ a Csebisev polinomok együtthatóit. N≤30 /bemenő/.

Irodalom:

[1] A.Ralston: Bevezetés a numerikus analizisbe. Müszaki Könyvkiadó, Budapest /1969/

[2] H.S.Wall: Analitic Theory of Continued Fractions, Van Nostrand, New York /1948/

[3] G.Hornecker:Chiffres I,/I958/ pp.I57-I69.

[4] G.Németh:Rept. of Central Research Institute for Physics, Budapest, KFKI-74-34.

FÜGGELÉK.

ILLUSZTRATIN PÉLDA FUTTATÁSOK.

1.

A FRACS1 program egy egyszerü alkalmazását mutatjuk be az alábbi példán.

Lánctört segitségével közelitő kifejezést határozunk meg az

$$f(x)=\frac{1}{x}\ln(1+x)=1-\frac{x}{2}+\frac{x^3}{3}-\frac{x^4}{4}+-\cdots\,, \quad 0\le x\le 1$$

függvény számára.

A feladatot megoldó program listája:

MAINPGM

```
      DOUBLE PRECISION C,D
      DIMENSION C(11),D(11)
      C(1)=1.ØD-Ø
      DO 1 I=2,11
      C(1)=-C(1)
    1 C(I)=C(1)/I
      CALL FRACS1(C,11,D)
      WRITE(3,2) C
    2 FORMAT(E35.16)
      STOP
      END
```

```
                               SCALAR MAP
SYMBOL     LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION
I               90

                               ARRAY MAP
SYMBOL     LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION
C               98         D             190

                               SUBPROGRAMS CALLED
SYMBOL     LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION
FRACS1         288         IBCOM#        28C

                               FORMAT STATEMENT MAP
SYMBOL     LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION        SYMBOL    LOCATION
    2          29C
                               STATEMENT LABEL MAP
 LOCATION   STA NUM      LABEL            LOCATION   STA NUM  LABEL  LOCATION    STA NUM  LABEL
  0001D8        3                          0001E0        4            0001F8         5
  0001FA        6          1               000238        7            000246         8
  00026C       10
   TOTAL MEMORY REQUIREMENTS 00027A BYTES

   HIGHEST SEVERITY LEVEL OF ERRORS FOR THIS MODULE WAS 2
```

/A listán szerepel még a FRACS1 szubrutin listája, ezt elhagyjuk az egyszerüség kedvéért./

```
                   DISK LINKAGE EDITOR DIAGNOSTIC OF INPUT

                        ACTION TAKEN      MAP
                        LIST    AUTOLINK    ILFIBCOM
                        LIST    AUTOLINK    ILFADCON
                        LIST    AUTOLINK    ILFFINT
                        LIST    AUTOLINK    ILFFIOCS
                        LIST    AUTOLINK    IJJCPD1
                        LIST    AUTOLINK    ILFUNTAB
                        LIST    ENTRY
```

```
16/10/74 PHASE      XFR-AD  LOCORE  HICORE  DSK-AD    ESD TYPE     LABEL      LOADED   REL-FR

         PHASE***   002800  002800  00657F  41 01 1   CSECT        MAINPGM    002800   002800

                                                      CSECT        FRACS1     002A80   002A80

                                                      CSECT        ILFIBCOM   002DA8   002DA8
                                                         ENTRY     IBCOM#     002DA8
                                                         ENTRY     READSW     003C38
                                                      *  ENTRY     OPSYS      003ADC
                                                         ENTRY     INTSW      003C3A
                                                         ENTRY     PDPAR      003A50
                                                         ENTRY     DUMPSW#    00390A
                                                      *  ENTRY     IJTINTSW   003C3A
                                                         ENTRY     IOSWF      002E99

                                                      CSECT        ILFFINT    004E68   004E68
                                                         ENTRY     SAVERR     005348

                                                      CSECT        ILFADCON   003D88   003D88
                                                         ENTRY     ILFFCVEO   004872
                                                         ENTRY     ILFFCVLO   00400A
                                                         ENTRY     ILFFCVIO   004348
                                                         ENTRY     ILFFCVCO   004A3C
                                                         ENTRY     ILFFCVAO   003F7A
                                                         ENTRY     ILFFCVZO   003ED4
                                                         ENTRY     INT6SW     004E50

                                                      CSECT        ILFFIOCS   005438   005438
                                                         ENTRY     ILFFBORG   006004
                                                         ENTRY     ILFBFORG   006000
                                                         ENTRY     IJSYSLO    006100
                                                      *  ENTRY     ADIOCR#    006014
                                                      *  ENTRY     UBRSAVE    00673C

                                                      CSECT        ILFUNTAB   006430   006468

                                                      CSECT        IJJCPD1    006238   006288
                                                      *  ENTRY     IJJCPD1N   006238
                                                      *  ENTRY     IJJCPD3    006238

                                                      CSECT        IJ2L0005   006248   005438

* UNREFERENCED SYMBOLS
```

A futtatás eredménye:

k	α_k
0	0.1000000000000000D 01
1	0.1000000000000000D 01
2	0.5000000000000000D 00
3	0.8333333333333333D-01
4	0.1388888888888889D-01
5	0.2314814814814836D-03
6	0.9645061728395152D-06
7	0.4784256839719558D-10
8	0.1318412921553038D-16
9	0.1491694666203018D-27
10	-0.1400009363643093D-44

ahol az α_k - k az

$$R = \cfrac{\alpha_1}{\alpha_0 + \cfrac{\alpha_2 x}{\alpha_1 + \cfrac{\alpha_3 x}{\alpha_2 + \dots}}},$$

lánctört együtthatói.

Látható, hogy az első hat együttható az alábbi

$$\alpha_0 = 1,$$
$$\alpha_1 = 1,$$
$$\alpha_2 = 1/2,$$
$$\alpha_3 = 1/12,$$
$$\alpha_4 = 1/72,$$
$$\alpha_5 = 1/4320.$$

Rövid számolással belátható, hogy

$$R_1(x) = \cfrac{1}{1 + \cfrac{x/2}{1 + \cfrac{x/12}{1/2}}} = \frac{6+x}{6+4x}.$$

Hasonlóan

$$R_2(x)=\cfrac{1}{1+\cfrac{x/2}{1+\cfrac{x/12}{\frac{1}{2}+\cfrac{x/12}{\frac{1}{12}+\cfrac{x/4320}{1/72}}}}}=\frac{30+21x+x^2}{30+36x+9x^2}.$$

Vizsgáljuk meg a közelitések pontosságát az x=1 helyen.

Mivel

$$f(1)=\ln 2=0.69315,$$

és

$$R_1(1)=0.7,$$

az eltérés $7\cdot10^{-3}$.

Hasonlóan

$$R_2(1)=0.69333$$

és az eltérés $1{,}8\cdot10^{-4}$.

Megjegyezzük, hogy a durvábbik $R_1(x)$ közelitéssel elért pontosságot az $\frac{1}{x}\ln(1+x)$ függvény sorával számolva csak több-száz tag figyelembevételével tudnánk elérni.

2.

Ebben a példában a CSPAMK szubrutin alkalmazásával mutatjuk be az $f(x)=e^{x}$ függvény racionális közelitésének meghatározását.

A program listát vázlatosan adjuk meg /elhagyva belőle a szubrutinok listázását és a forditó program közleményeit/.

A // jellel bevezetett utasitások a DOS operációs rendszer vezérlő utasitásai.

```
// JOB R.S812                                          12.12.22   DATE 29/07/74
// OPTION LINK
// EXEC FFORTRAN

DOS FORTRAN IV   KFKI

 OPTIONS IN EFFECT
 -----------------

 LOAD   =4

 DECK   NO

 LIST   YES

 LISTX  NO

 EBCDIC

            DOUBLE PRECISION A1,A2,A3,A4,B1,B2,B3,X,C,FK,E1,E2,A,B,W,EPS
            DIMENSION C(32),E1(8),E2(8),A(8),B(8),D(8),W(8)
            CALL EGYUTT(25)
     3      READ(1,1) A1,A2,A3,K
            READ(1,1) A4,B1,B2,N
            READ(1,1) B3,X,FK,L
     1      FORMAT(3D22.16,I3)
            X=(((A1*A2)*X/B1/B2)*A3*A4)/B3
            CALL HIPE43(A1,A2,A3,A4,B1,B2,B3,X,K,N,C,L,FK)
            C(1)=2*C(1)
            DO 2 I=2,6
            M=3*I+2
            CALL CSPAHK(C,M,E1,I+1,E2,I+1,EPS,A,B)
            E1(I+2)=EPS
            E2(I+2)=0.0D-0
            CALL KIIR(C,E1,E2,I+2)
            CALL RENDZ3(E1,I+1,E2,I+1,1,W)
            CALL KIIR(C,E1,E2,I+1)
     2      CONTINUE
            GO TO 3
            END
```

A programlistán itt következnek a felhasznált szubrutinok listái: CSPAMK, CSETAY, EGYUTT, RENDZ3, HIPE43, MEGOLD /elhagyjuk őket/.

Szerepel még egy /PRAKA-hoz nem tartozó, felhasználói/ szubrutin a kiíráshoz:

```
      SUBROUTINE KIIR(A,B,C,N)
      DOUBLE PRECISION A(N),B(N),C(N)
      DO 2 I=1,N
      WRITE(3,4) A(I),B(I),C(I)
    2 CONTINUE
    4 FORMAT(3E35.16)
      RETURN
      END
// EXEC LNKEDT

// EXEC
```

A futtatás eredményei:

/ $f(x)=e^x$ Csebisev - Padé közelitései/.

	0.1290070540898370D 01	0.6365697825132393D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.1302372218762056D 00	0.2471752614327729D 00
	0.3872411541932658D-01	0.6240389819138536D-02	0.1022449332277410D-01
n=2	-0.3208683015139925D-02	$\lambda=$ $\boxed{-0.1620298801772689D-05}$	0.0
	0.1290070540898370D 01	0.9999975910607462D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.4592070179241485D 00	0.5406557382532178D 00
	0.3872411541932658D-01	0.6542581586012797D-01	0.1071961587322300D 00
	0.1290070540898370D 01	0.6065474539136282D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.1502983898498706D 00	0.2475754593913379D 00
	0.3872411541932658D-01	0.7192640677591225D-02	0.1232562022536956D-01
	-0.3208683015139925D-02	-0.1550647909540024D-03	0.2558324221668761D-03
n=3	0.1999192377554991D-03	$\lambda=$ $\boxed{0.7232015347737916D-09}$	0.0
	0.1290070540898370D 01	0.1000000000072841D 01	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.4752729455984867D 00	0.5247271636561299D 00
	0.3872411541932658D-01	0.6819245987943984D-01	0.1129177841229698D 00
	-0.3208683015139925D-02	-0.6524146910140347D-02	0.1070856539433628D-01

	0.1290070540898300D 01	0.6065399609382952D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.1560335498509168D 00	0.2477331706830336D 00
	0.3870411541932658D-01	0.6730370329099964D-02	0.1322445836637138D-01
	-0.3208683015133925D-02	-0.2227437490424338D-03	0.3664767847861519D-03
	0.1999192377554991D-03	0.2781082291671816D-05	0.4570646673369762D-05
n=4	-0.9975211042940343D-05	λ= $\boxed{-0.1793638694875057D-12}$	0.0
	0.1290070540898300D 01	0.9999999999999734D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.4622623160159822D 00	0.5177175839397497D 00
	0.3870411541932658D-01	0.9251697770298784D-01	0.1162366621964308D 00
	-0.3208683015133925D-02	-0.1324658289917649D-01	0.1379789279374057D-01
	0.1999192377554991D-03	0.4653665670675393D-03	0.7646324836321318D-03
	0.1290070540898300D 01	0.6065365923674115D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.1503561505732407D 00	0.2478157147130544D 00
	0.3870411541932658D-01	0.8329434732495096D-02	0.1372320297733575D-01
	-0.3208683015133925D-02	-0.2599274046310796D-03	0.4286253574262426D-03
	0.1999192377554991D-03	0.4637609414403064D-05	0.7631645306625376D-05
n=5	-0.9975211042940343D-05	-0.3562480152036653D-07	0.6350864667128685D-07
	0.4150168956922782D-06	λ= $\boxed{0.2830837607696595D-16}$	0.0
	0.1290070540898300D 01	0.1000000000000000D 01	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.4861967463361278D 00	0.5138032536636838D 00
	0.3870411541932658D-01	0.1243386475645498D 00	0.1181419912280659D 00
	-0.3208683015133925D-02	-0.1247332328852872D-01	0.1543361778112973D-01
	0.1999192377554991D-03	0.8463535417106895D-03	0.1169934595110656D-02
	-0.9975211042940343D-05	-0.2363440704157626D-04	0.4247286415274643D-04
	0.1290070540898300D 01	0.6065347675268989D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.1503719523942761D 00	0.2478664138144726D 00
	0.3870411541932658D-01	0.8519996611047168D-02	0.1424031562254156D-01
	-0.3208683015133925D-02	-0.2836024707950564D-03	0.4672061136000631D-03
	0.1999192377554991D-03	0.5903419791162302D-05	0.9721389667566648D-05
	-0.9975211042940343D-05	-0.7375136822972088D-07	0.1213867737611845D-06
n=6	0.4150168956922782D-06	0.4368235626860715D-09	0.7217555297053911D-09
	-0.1480552232566769D-07	λ= $\boxed{-0.3993123296618263D-25}$	0.0
	0.1290070540898300D 01	0.9999999999999997D 00	0.1000000000000000D 01
	-0.3128416063697435D 00	-0.4886945106183728D 00	0.5113054893816277D 00
	0.3870411541932658D-01	0.1286652301758963D 00	0.1193707195525150D 00
	-0.3208683015133925D-02	-0.1393561671555412D-01	0.1644902481281467D-01
	0.1999192377554991D-03	0.1114084474793197D-02	0.1428664408311009D-02
	-0.9975211042940343D-05	-0.5283526614254668D-04	0.7537460344181439D-04
	0.4150168956922782D-06	0.1173676184195355D-05	0.1930421661073649D-05

A számitás eredménye n=2 esetén az alábbi:

$$R_2(x) = \frac{0.606569.. - 0.150237... * T_1^*(x) + 0.006240.. * T_2^*(x)}{1.0 \quad + 0.247175... * T_1^*(x) + 0.010224... * T_2^*(x)} =$$

$$= \frac{0.999997... - 0.459207... * X + 0.065425... * X^2}{1.0 \quad + 0.540665... * X + 0.107196... * X^2},$$

$$\lambda = 1{,}62 . 10^{-6},$$

azaz az eredmény ki van nyomtatva Csebisev $T_k^*(x)$ polinomok szerint rendezett alakban és a szokásosabb algebrai polinomok szerint rendezett alakban.

A λ szám a tört Csebisev - Padé módszerrel számitott hibája.

Összehasonlitásképpen megadjuk e^x másodfoku polinomokkal való legjobb racionális közelitését:

$$R_2^*(x) = \frac{0.99999836 - 0.46071168\, X + 0.06599183\, X^2}{1.0 \quad + 0.53918916\, X + 0.10612105\, X^2},$$

e közelités hibája:

$$|\lambda^*| = 1{,}64 . 10^{-6}.$$

Látható, hogy gyakorlati célokra /a kicsiny eltérés miatt/ a Csebisev - Padé közelités olyan jó mint a legjobb egyenletes megközelités.

Kiadja a Központi Fizikai Kutató Intézet
Felelős kiadó: Varga László, a KFKI Számítástechnikai Tudományos Tanácsának szekcióelnöke
Szakmai lektor: Hegedüs Csaba
Példányszám: 145 Törzsszám: 75-045
Készült a KFKI sokszorositó üzemében
Budapest, 1975. január hó